

# Настройка сервера Minecraft на CentOS

**Minecraft** - Компьютерная инди-игра в жанре песочницы, разработанная шведским программистом Маркусом Перссоном и выпущенная его компанией **Mojang AB**. В этой статье мы создадим сервер для игры в Minecraft, рассмотрим решение некоторых типовых ошибок, настроим лаунчер на клиенте и сделаем рассылку друзьям с приглашением поиграть!

Принцип работы системы Minecraft: есть сервер с миром и локациями, к нему подключается клиент с игроком. Сервер Minecraft будет создан на базовой оригинальной сборке от разработчиков, без модификаций и прочих расширений. На момент написания статьи последняя версия игры Minecraft была **1.14.4**. Это стандартный, он же официальный чистый сервер майнкрафт для **1.14.4**, в нем нет плагинов или модов, данный сервер нужен для игры с друзьями в классический майнкрафт подобно одиночной игре.







Игровой сервер Minecraft имеет **порт 25565**, который можно легко поменять его в файлах конфигурации. К серверу подключимся с помощью **TL-лаунчера** для Minecraft.

## Настройка сервера Minecraft в CentOS

### 1. Подготовка сервера

Для создания сервера Minecraft достаточно установить сервер CentOS 7 в Минимальной конфигурации. Всё остальное доустановим в ходе настройки сервера. Установим текстовый редактор **mc** и **net-tools** пакет для определения **IP-адреса** сервера:

```
yum -y install mc net-tools
```

Обновим все компоненты CentOS 7 и их зависимости до последних версий в репозиториях:

```
yum -y upgrade
```

Убедимся, что наш сервер получил **IP адрес** по **DHCP** с источника и имеет свой **IP-адрес**:

ifconfig

```
[root@probookcentos7 ~]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.10.132 netmask 255.255.255.0 broadcast 192.168.10.255
    inet6 fe80::fd86:8f1e:65e9:6ba prefixlen 64 scopeid 0x20<link>
    ether 12:34:56:ff:ff:ff txqueuelen 1000 (Ethernet)
    RX packets 389735 bytes 339155458 (323.4 MiB)
    RX errors 0 dropped 21 overruns 0 frame 0
    TX packets 116529 bytes 11514050 (10.9 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Наш сервер получил IP-адрес **192.168.10.132**. Запомним его. Он пригодится нам позднее для настройки файла конфигурации игрового сервера Minecraft.

### 3. Установка java на сервер

Убедимся установлена ли у нас **java** вообще или нужно ли ее переустановить на более новую версию?

java -version

```
[root@probookcentos7 /]# java -version
java version "1.6.0_41"
OpenJDK Runtime Environment (IcedTea6 1.13.13) (rhel-1.13.13.1.el7_3-x86_64)
OpenJDK 64-Bit Server VM (build 23.41-b41, mixed mode)
[root@probookcentos7 /]#
```

В данном случае в нашей **CentOS 7** уже стоит версия **1.6.0**, но она уже устарела для версии игры **1.14.4** и нам нужно будет установить более свежую версию для корректной работы нашего сервера Minecraft.

Посмотрим какие версии **java** вообще имеются в доступных репозиториях:

```
[root@probookcentos7 ~]# yum search java
Загружены модули: fastestmirror
Loading mirror speeds from cached hostfile
* base: mirrors.datahouse.ru
* extras: mirrors.datahouse.ru
* updates: mirrors.datahouse.ru
===== N/S matched: java ==
```

В списке будет много-много всего, что содержит слово **java**, но нас интересуют пакеты такого вида:

```

java-1.8.0-openjdk.i686 : OpenJDK Runtime Environment 8
java-1.8.0-openjdk.x86_64 : OpenJDK Runtime Environment 8
java-1.8.0-openjdk-accessibility.i686 : OpenJDK accessibility connector
java-1.8.0-openjdk-accessibility.x86_64 : OpenJDK accessibility connector
java-1.8.0-openjdk-accessibility-debug.i686 : OpenJDK 8 accessibility connector for packages with debug on
java-1.8.0-openjdk-accessibility-debug.x86_64 : OpenJDK 8 accessibility connector for packages with debug on
java-1.8.0-openjdk-debug.i686 : OpenJDK Runtime Environment 8 with full debug on
java-1.8.0-openjdk-debug.x86_64 : OpenJDK Runtime Environment 8 with full debug on
java-1.8.0-openjdk-demo.i686 : OpenJDK Demos 8
java-1.8.0-openjdk-demo.x86_64 : OpenJDK Demos 8
java-1.8.0-openjdk-demo-debug.i686 : OpenJDK Demos 8 with full debug on
java-1.8.0-openjdk-demo-debug.x86_64 : OpenJDK Demos 8 with full debug on
java-1.8.0-openjdk-devel.i686 : OpenJDK Development Environment 8
java-1.8.0-openjdk-devel.x86_64 : OpenJDK Development Environment 8
java-1.8.0-openjdk-devel-debug.i686 : OpenJDK Development Environment 8 with full debug on
java-1.8.0-openjdk-devel-debug.x86_64 : OpenJDK Development Environment 8 with full debug on
java-1.8.0-openjdk-headless.i686 : OpenJDK Headless Runtime Environment 8
java-1.8.0-openjdk-headless.x86_64 : OpenJDK Headless Runtime Environment 8
java-1.8.0-openjdk-headless-debug.i686 : OpenJDK Runtime Environment with full debug on
java-1.8.0-openjdk-headless-debug.x86_64 : OpenJDK Runtime Environment with full debug on
java-1.8.0-openjdk-javadoc.noarch : OpenJDK 8 API documentation
java-1.8.0-openjdk-javadoc-debug.noarch : OpenJDK 8 API documentation for packages with debug on
java-1.8.0-openjdk-javadoc-zip.noarch : OpenJDK 8 API documentation compressed in a single archive
java-1.8.0-openjdk-javadoc-zip-debug.noarch : OpenJDK 8 API documentation compressed in a single archive for packages with debug on
java-1.8.0-openjdk-src.i686 : OpenJDK Source Bundle 8
java-1.8.0-openjdk-src.x86_64 : OpenJDK Source Bundle 8
java-1.8.0-openjdk-src-debug.i686 : OpenJDK Source Bundle 8 for packages with debug on
java-1.8.0-openjdk-src-debug.x86_64 : OpenJDK Source Bundle 8 for packages with debug on

```

Ставим все пакеты:

```
yum -y install java-1.8.0*
```

Проверим версию **java**:

```
java -version
```

```

[root@probookcentos7 ~]# java -version
openjdk version "1.8.0_232"
OpenJDK Runtime Environment (build 1.8.0_232-b09)
OpenJDK 64-Bit Server VM (build 25.232-b09, mixed mode)
[root@probookcentos7 ~]# █

```

Порядок! Далее устанавливаем **screen**, для того чтоб выполнить запуск **сервера MineCraft** и он продолжил работу после того как выйдем из консоли **VPS**:

```
yum -y install screen
```

Запуски **сервера Minecraft** будем делать в **screen** окне.

### 3. Как работать в screen

Создание окна с уникальным именем:

```
screen -S "имя_окна"
```

Если вдруг запутаемся и потеряемся в screen окнах, всегда можно набрать специальную команду и посмотреть меню окон, что у нас открыто вообще:

```
screen -r
```

Затем ввести **ID окна** или его имя в кавычках, данное при создании окна и переключиться на окно "**имя\_окна**":

```
screen -r "имя_окна"
```

Еще можно учитывать, что все наши труды работают во всех созданных окнах в фоновом режиме, благодаря **screen**.

Нажмем одновременно клавиши «**Ctrl**» + «**a**», затем просто одну клавишу «**d**». **Screen** с нашими трудами свернётся в память, а мы выйдем на первичную консоль **CentOS 7**. Вы можете вернуться в меню окон **screen** это и снова открыть окно **screen**, введя в терминале **screen -r** и имя окна, которое дали ему при создании его.

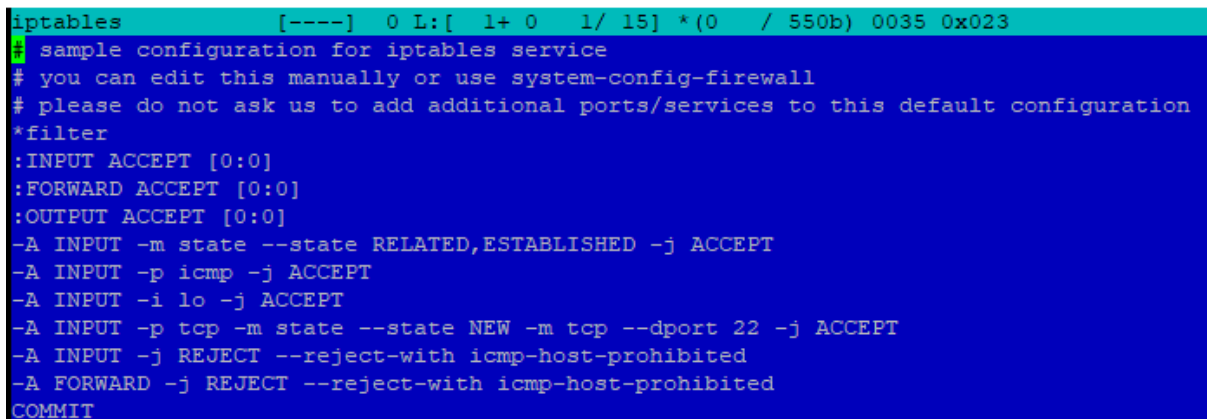
Вернуться обратно в **screen** рабочим процессом, всегда просто и легко с помощью этой же команды:

```
screen -r "имя_окна"
```

## 4. Настройка firewall

Для корректной работы сервера Minecraft нужно настроить правила **firewall** на сервере **CentOS 7**. В руководстве используется программа **iptables**. Скорее всего, что настройки в других **firewall** продуктах будут аналогичными. Будем считать, что на сервере уже успешно функционирует **iptables** и установлен текстовый редактор **mc**. Открываем файл конфигурации **iptables** текстовым редактором:

```
mcedit /etc/sysconfig/iptables
```



```
iptables      [-----] 0 L:[ 1+ 0  1/ 15] *(0  / 550b) 0035 0x023
# sample configuration for iptables service
# you can edit this manually or use system-config-firewall
# please do not ask us to add additional ports/services to this default configuration
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
```

И в середине массива строк, под разрешением работы с **портом 22**, добавляем строку точно такую же строку, но с **портом** игрового **сервера Minecraft** - это **порт 25565**:

```
-A INPUT -p tcp -m state --state NEW -m tcp --dport 25565 -j ACCEPT
```

Должно получиться так:

```

iptables      [----]  0 L:[  1+14  15/ 15] *(534 / 534b) <EOF>
# Generated by iptables-save v1.4.21 on Tue Nov 19 18:26:17 2019
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 25565 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT

```

Чтобы **iptables** это учел в работе, нужно его перезапустить:

```
systemctl restart iptables
```

Если после этой команды вышла ошибка:

```

[root@probookcentos7 ~]# systemctl restart iptables
Job for iptables.service failed because the control process exited with error code. See "systemctl status iptables.service" and "journalctl -xe" for details.

```

Внимательно проверяйте, что вы там написали. В вашей строке содержатся опечатки!

## 6. Установка сервера Minecraft

Для удобства запуска сервера Minecraft создаем папку **minecraft** в корневом каталоге сервера. Качаем хоть откуда файл с сервером Minecraft. Любым способом помещаем файл сервера Minecraft в этот каталог.

Получится:

```
/minecraft/minecraft_server.1.14.4.jar
```

Теперь начнем активацию сервера Minecraft. Запускаем **screen**:

```
screen -S "Minecraft server"
```

Запускаем **сервер Minecraft**:

```
java -Xmx1024M -Xms1024M -jar /minecraft/minecraft_server.1.14.4.jar nogui
```

Он надолго задумается и станет распаковывать своё содержимое в корневую папку системы **CentOS 7**, затем выдаст сообщение о том, что бы мы прочитали и приняли лицензионное соглашение по игре **Minecraft**.

```
[root@probookcentos7 ~]# java -Xmx1024M -Xms1024M -jar /minecraft/minecraft_server.1.14.4.jar nogui
[13:20:22] [main/ERROR]: Failed to load properties from file: server.properties
[13:20:23] [main/WARN]: Failed to load eula.txt
[13:20:23] [main/INFO]: You need to agree to the EULA in order to run the server. Go to eula.txt for more info.
[root@probookcentos7 ~]#
```

Нужно утвердительно ответить ему в файле **eula.txt**, что да, мы читали и согласны со всем написанным!

Наши труды работают в фоновом режиме, благодаря **screen**. Нажмем одновременно клавиши «**Ctrl**» + «**a**», затем просто одну клавишу «**d**». **Screen** с нашими трудами свернётся в память, а мы выйдем на первичную консоль **CentOS 7**. Вы можете вернуться в меню окон **screen** это и снова открыть окно **screen**, введя в терминале **screen -r** и имя окна, которое дали ему при создании его.

Вернемся обратно в screen рабочим процессом по серверу **Minecraft**, как только понадобится:

```
screen -r "Minecraft server"
```

Вышли и видим:

```
[detached from 15877.Minecraft server]
[root@probookcentos7 ~]#
```

Открываем текстовый файл **eula.txt** в текстовом редакторе, он распаковывается в домашнюю папку пользователя, под которым вы вошли в систему **CentOS 7**.

```
mcedit ~/.minecraft/eula.txt
```

или как-то так

```
mcedit ~/.minecraft/eula.txt
```

```
eula.txt  [----] 0 L:[ 1+ 3  4/  4] *(182 / 182b) <EOF>
#By changing the setting below to TRUE you are indicating your agreement to our EULA (https://account.mojang.com/documents/minecraft_eula).
#Wed Nov 20 13:20:23 YEKT 2019
eula=false
```

Меняем false на true значение по смыслу.

```
eula.txt  [-M--] 0 L:[ 1+ 3  4/  4] *(181 / 181b) <EOF>
#By changing the setting below to TRUE you are indicating your agreement to our EULA (https://account.mojang.com/documents/minecraft_eula).
#Wed Nov 20 13:20:23 YEKT 2019
eula=true
```

Сохраняем файл. Теперь нужно отключить проверку лицензий на Minecraft на клиентах сервера Minecraft и добавить в него IP-адрес сервера CentOS 7. Эти параметры находится в файле конфигурации сервера Minecraft **server.properties**, который распаковывается всегда рядом с **eula.txt**:

```
mcedit ~/.minecraft/server.properties
```

Ищем строки '**server-ip=**' (он пустой параметр содержит) и '**online-mode=true**' (активирована проверка), которые содержат эти параметры и заменяем их на вот эти значения:

```
server-ip=192.168.10.132
```

```
online-mode=false
```

Сохраняем файл конфигурации. Так же этот файл содержит множество других параметров, которые непосредственно влияют на правила игры, но их описание можно легко найти на просторах Интернета и тратить на них место в статье нет смысла.

Продолжим запускать сервер Minecraft:

```
screen -r "Minecraft server"
```

Снова запускаем **java** файл **minecraft\_server.1.14.4.jar**:

```
java -Xmx1024M -Xms1024M -jar /minecraft/minecraft_server.1.14.4.jar nogui
```

Minecraft требует наличия **1 ГБ** оперативной памяти для загрузки сервера, этого вполне хватит. Но всё равно рекомендуется увеличить объем оперативной памяти для повышения производительности! Для этого замените **1024** в параметрах **-Xmx** и **-Xms** на **2048 (2 ГБ)** и так далее. Подробнее параметры запуска так же можно почитать в Интернете.

Для **32-bit** системы префикс **-d32**, для **64-bit** системы префикс **-d64**. Вариант щедрого запуска сервера на 64-bit с 4 Gb оперативной памяти:

```
java -d64 -Xmx496M -Xms496M -jar /minecraft/minecraft_server.1.14.4.jar nogui
```

Файл долго-долго подумает, подзависнет и резко проявится вот таким окончанием работы процесса:

```
[13:54:05] [Server-Worker-1/INFO]: Preparing spawn area: 86%
[13:54:06] [Server-Worker-1/INFO]: Preparing spawn area: 88%
[13:54:06] [Server-Worker-1/INFO]: Preparing spawn area: 89%
[13:54:07] [Server-Worker-1/INFO]: Preparing spawn area: 90%
[13:54:07] [Server-Worker-1/INFO]: Preparing spawn area: 92%
[13:54:08] [Server-Worker-1/INFO]: Preparing spawn area: 92%
[13:54:08] [Server-Worker-1/INFO]: Preparing spawn area: 93%
[13:54:09] [Server-Worker-1/INFO]: Preparing spawn area: 96%
[13:54:09] [Server-Worker-1/INFO]: Preparing spawn area: 98%
[13:54:10] [Server thread/INFO]: Time elapsed: 49921 ms
[13:54:10] [Server thread/INFO]: Done (100.786s)! For help, type "help"
```

Это означает, что в оперативной памяти сервера **CentOS 7** был создан и заработал сервер Minecraft!

Наши труды работают в фоновом режиме, благодаря **screen**. Нажмем одновременно клавиши «**Ctrl**» + «**a**», затем просто одну клавишу «**d**». **Screen** с нашими трудами свернётся в память, а мы выйдем на первичную консоль **CentOS 7**. Вы можете вернуться в меню окон **screen** это и снова открыть окно **screen**, введя в терминале **screen -r** и имя окна, которое дали ему при создании его.

Вернемся обратно в screen рабочим процессом по **серверу Minecraft** если нас это снова потребуется:

```
screen -r "Minecraft server"
```

Пока, выйдем из этого screen и проверим что **сервер Minecraft** в оперативной памяти. Вышли и видим:

```
[detached from 15877.Minecraft server]
[root@probookcentos7 ~]#
```

Просканируем оперативную память на наличие процессов в ней по открытым портам:

```
netstat -ntulp
```

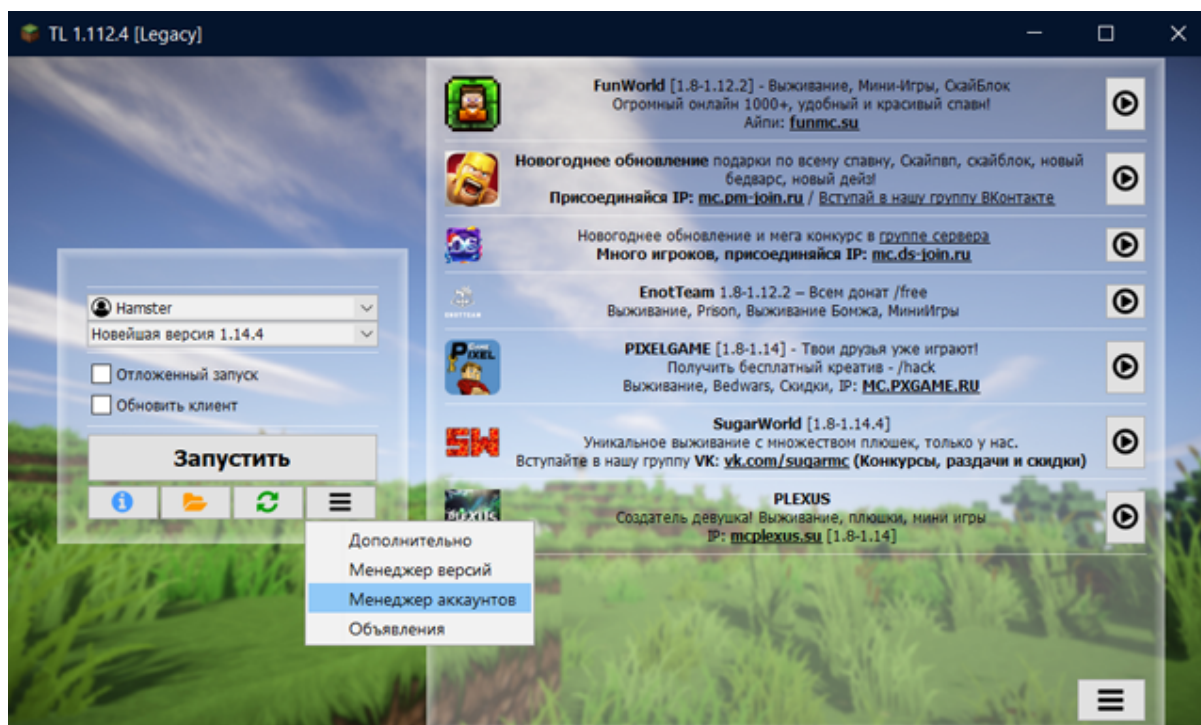
```
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      1015/master
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      931/sshd
tcp6       0      0 :::1:25                  :::*                    LISTEN      1015/master
tcp6       0      0 192.168.10.132:25565    :::*                    LISTEN      7828/java
tcp6       0      0 :::22                    :::*                    LISTEN      931/sshd
udp        0      0 0.0.0.0:68              0.0.0.0:*               1477/dhclient
udp        0      0 127.0.0.1:323           0.0.0.0:*               720/chronyd
udp6       0      0 :::1:323                 :::*                    720/chronyd
[root@probookcentos7 ~]#
```

Как видно наш процесс **7828/java** висит в оперативной памяти имеет **IP-адрес 192.168.10.132** и исправно слушает **25565 порт**! Подключение к серверу будет происходить по адресу **192.168.10.132:25565**.

## 6. Настройка клиента Minecraft

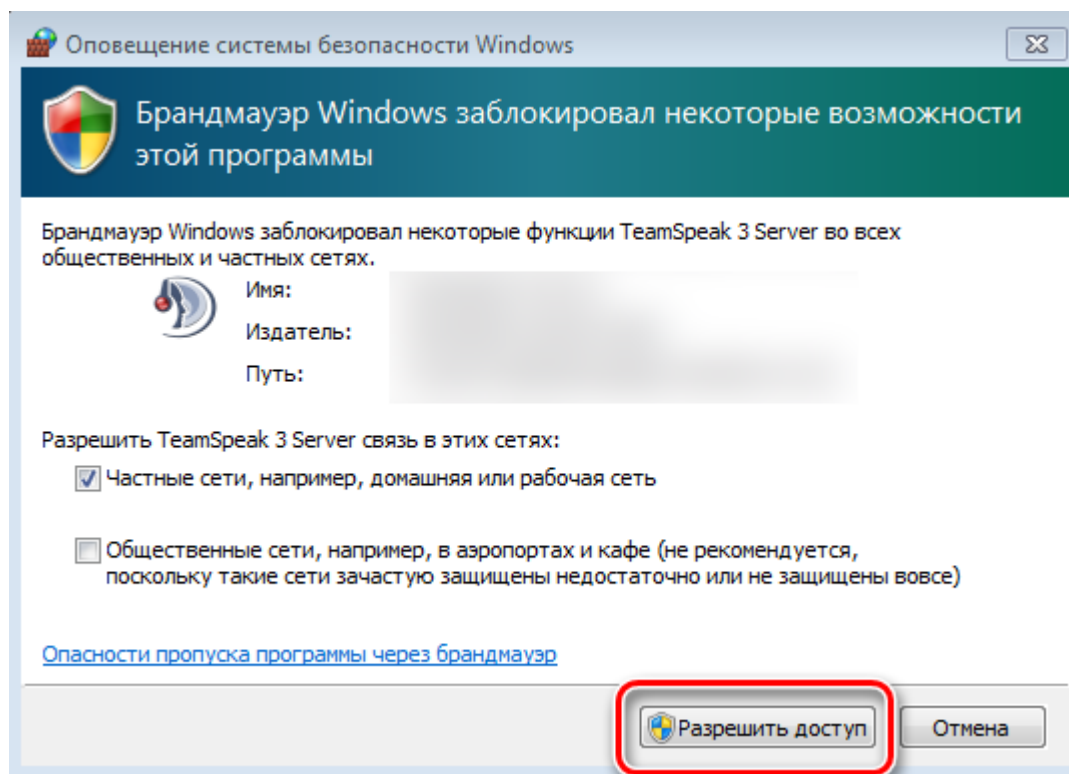
В начале скачаем и установим свежайшую версию **java** под именно вашу операционную систему. **Java** on-line установщик [\[Скачать 2 Мб\]](#). Установите его в вашу систему. Если предложит удалить старую версию, то согласитесь и установите новую.

В качестве лаунчера игры Minecraft будет использоваться TL-лаунчер. В сообществе всегда можно скачать последнюю версию данной программы. Для запуска игры подойдет любой другой лаунчер с идентичным клиентом сборки Minecraft, как на игровом сервере Minecraft. Клиент Minecraft (TL-лаунчер) [\[Скачать 2,2 Мб\]](#). Скачаем TL-лаунчер в любую папку в Windows и запустим его:

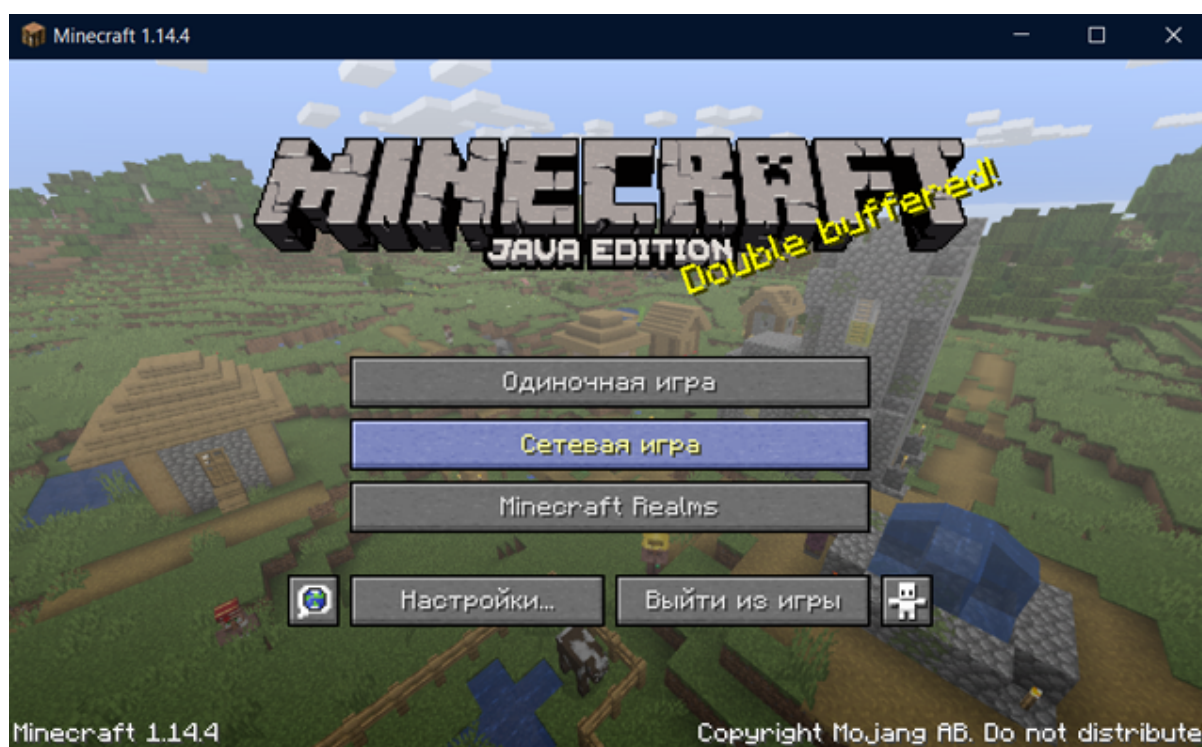


В лаунчере нужно создать имя учетной записи, выбрать в окошке запуска **Имя учетной записи** и **Версию клиента** игры. Так как у нас сервер Minecraft создан на версии **1.14.4** без модификаций, то его и выбираем из списка. Если лаунчер предложит докачать какие-то компоненты для игры, то соглашайтесь. Он сделает это сам автоматически. Жмем **Установить**, потом **Запустить**.

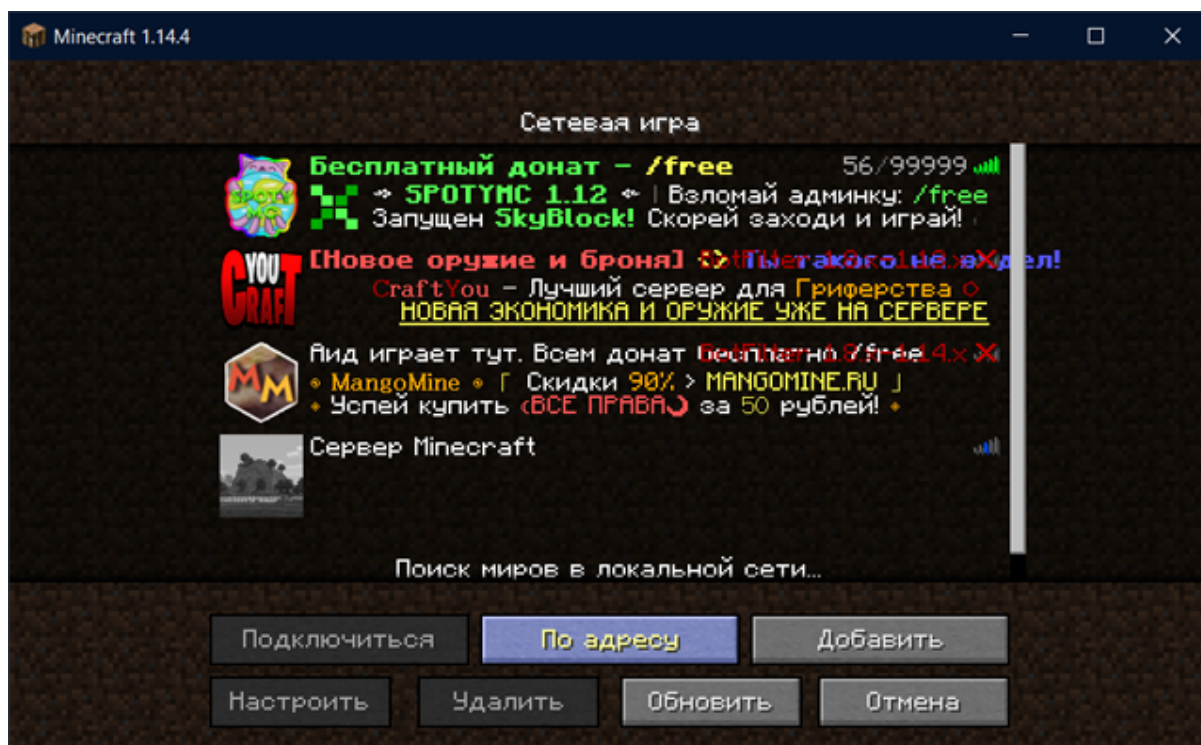
Возможно, вам потребуется добавить сервер Minecraft в разрешенные приложения в брандмауэре. Ставим все галочки на разрешение и подтверждаем согласие.



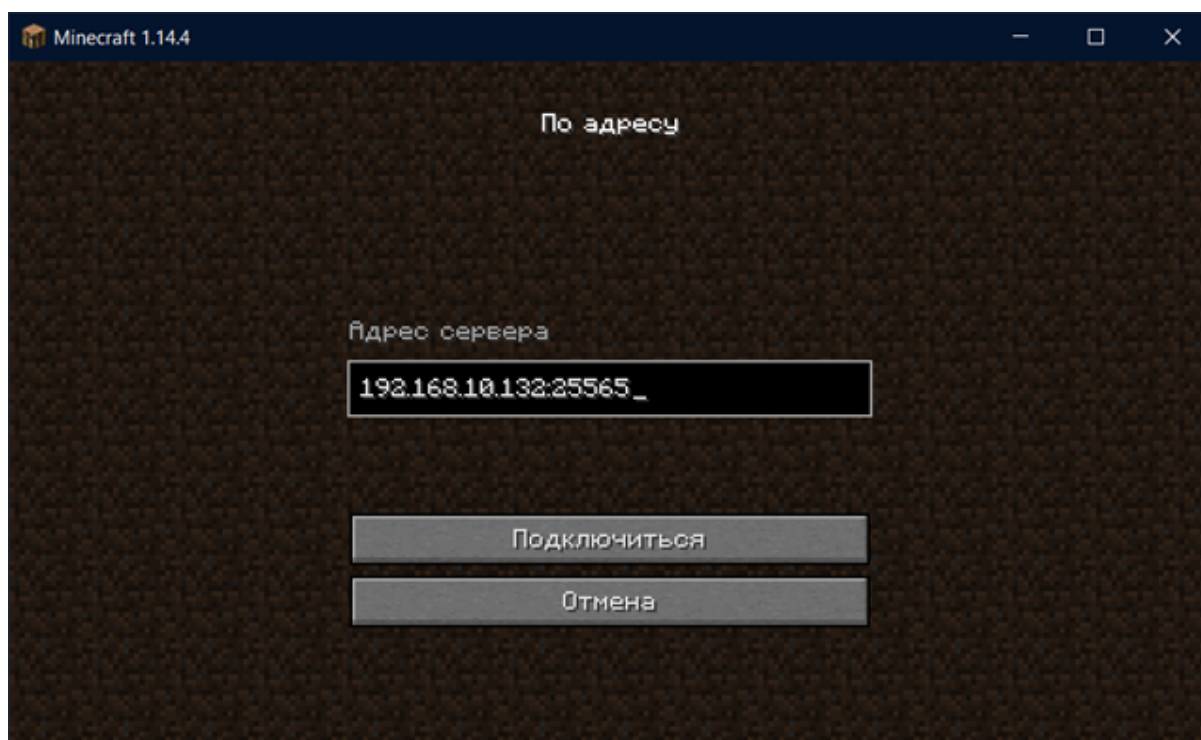
Игра запускается... Выбираем **Сетевая игра**.



Подключение **По адресу**.



Вводим IP-адрес **192.168.10.132:25565** нашего игрового сервера **Minecraft** и нажимаем **Подключиться**.



Если всё было сделано правильно, то игрок окажется в игровом мире **Minecraft**. Приятной игры!

## 7. Проброс порта в Интернет

Для совместной игры с друзьями в Интернете, нужно сделать проброс порта любым способом. Наиболее удобный способ сделать это через роутер с белым IP-адресом.

## Команды сервера Minecraft

### Назначение Администратора сервера

Для того, чтобы дать администраторские права на учетную запись в игровом мире, нужно зайти пользователем на сервер Minecraft, отметить там в игре, далее примерить команду на сервере **CentOS 7** в консоли запущенной панели управления сервера Minecraft:

```
/op user_name
```

Пример с пользователем **Hamster**:

```
[17:36:10] [Server-Worker-1/INFO]: Preparing spawn area: 96%  
[17:36:10] [Server-Worker-1/INFO]: Preparing spawn area: 98%  
[17:36:11] [Server thread/INFO]: Time elapsed: 43499 ms  
[17:36:11] [Server thread/INFO]: Done (49.828s)! For help, type "help"  
/op Hamster  
[17:37:44] [Server thread/INFO]: Made Hamster a server operator
```

### Остановка сервера

Иногда возникает потребность выключить сервер с **CentOS 7**, но у нас там работает сервер Minecraft! Так просто выключить нельзя. Перед выключением очень необходимо корректно завершить работу Minecraft, что бы сохранить всю игровую локацию и не повредить файлы с игровым миром. Прописываем команду:

```
/stop
```

```
/stop  
[10:36:10] [Server thread/INFO]: Stopping the server  
[10:36:10] [Server thread/INFO]: Stopping server  
[10:36:10] [Server thread/INFO]: Saving players  
[10:36:10] [Server thread/INFO]: Saving worlds  
[10:36:10] [Server thread/INFO]: Saving chunks for level 'world'/minecraft:overworld  
[10:36:12] [Server thread/INFO]: ThreadedAnvilChunkStorage (world): All chunks are saved  
[10:36:12] [Server thread/INFO]: Saving chunks for level 'world'/minecraft:the_end  
[10:36:12] [Server thread/INFO]: ThreadedAnvilChunkStorage (DIM1): All chunks are saved  
[10:36:12] [Server thread/INFO]: Saving chunks for level 'world'/minecraft:the_nether  
[10:36:12] [Server thread/INFO]: ThreadedAnvilChunkStorage (DIM-1): All chunks are saved  
[10:36:12] [Server thread/INFO]: ThreadedAnvilChunkStorage (world): All chunks are saved  
[10:36:12] [Server thread/INFO]: ThreadedAnvilChunkStorage (DIM1): All chunks are saved  
[10:36:12] [Server thread/INFO]: ThreadedAnvilChunkStorage (DIM-1): All chunks are saved  
[root@probookcentos7 ~]#
```

## Что делать, если сервер Minecraft завис?

Просканируем оперативную память на наличие процессов в ней по открытым портам:

```
netstat -ntulp
```

```
[root@probookcentos7 ~]# netstat -ntulp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      1017/master
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      934/sshd
tcp6       0      0 :::1:25                 :::*                    LISTEN      1017/master
tcp6       0      0 192.168.10.132:25565    :::*                    LISTEN      1386/java
tcp6       0      0 :::22                   :::*                    LISTEN      934/sshd
udp        0      0 0.0.0.0:68              0.0.0.0:*               1282/dhclient
udp        0      0 127.0.0.1:323           0.0.0.0:*               715/chronyd
udp6       0      0 :::1:323                 :::*                    715/chronyd
```

Что мы видим? Ага, строка с **IP-адресом 192.168.10.132** открыт **порт 25565** - это и есть наш процесс с сервером Minecraft. Нас интересует номер процесса в последнем столбце. Это будет значение **1386/java**. Принудительно закроем зависший процесс сервера. Чтобы послать сигнал **SIGKILL** (он имеет **номер 9**) процессу **1386**, выполните в командой строке:

```
kill -9 1386
```

Вообще команда **kill** предназначена для посылки сигнала процессу. По умолчанию, если мы не указываем какой сигнал посылать, посылается сигнал **SIGTERM** (от слова **termination** — завершение). **SIGTERM** указывает процессу на то, что необходимо завершиться. Каждый сигнал имеет свой номер. **SIGTERM** имеет номер **15**. Список всех сигналов (и их номеров), которые может послать команда **kill**, можно вывести, выполнив **kill -l**.

Применим команду и снова просканируем оперативную память на наличие процессов в ней по открытым портам:

```
netstat -ntulp
```

```
[root@probookcentos7 ~]# kill -9 1386
[root@probookcentos7 ~]# netstat -ntulp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      1017/master
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      934/sshd
tcp6       0      0 :::1:25                 :::*                    LISTEN      1017/master
tcp6       0      0 :::22                   :::*                    LISTEN      934/sshd
udp        0      0 0.0.0.0:68              0.0.0.0:*               1282/dhclient
udp        0      0 127.0.0.1:323           0.0.0.0:*               715/chronyd
udp6       0      0 :::1:323                 :::*                    715/chronyd
[root@probookcentos7 ~]#
```

Как видно, процесс с сервером Minecraft в оперативной памяти сервера **CentOS 7** отсутствует. Теперь можно снова запустить сервер Minecraft по инструкции выше.

Если что-то идет не так вообще, то сотрите все файлы, которые создались при установке и заново выполните действия в этой инструкции. Ошибки должны исчезнуть.