

Creating a Minecraft Server - Tablets

SLQ Wiki Fabrication Lab 2026/01/15 21:29

Creating a Minecraft Server - Tablets

This version of Minecraft is for Tablet clients (iPad, Android, xBox) and will not work with Windows/Mac. Instructions on [creating a server for desktop clients is available here](#).

This tutorial is for creating a Minecraft server using PocketMine on Ubuntu 18. When choosing your server, be sure that it has at a minimum of 1GB of RAM. The more players you want to host, the more RAM you will require. As a guide, 1GB is a good amount for 10 players.
We are assuming you have access to root access for this tutorial.

Make a new user for Minecraft

Before we get started, lets make a new user just the run the server - this is good security practice in case we want to run our server public.

```
sudo adduser mcuser
```

You'll be prompted to add a password. Lets add the user to sudo group (superuser)

```
sudo usermod -aG sudo mcuser
```

And switch to the user

```
su - mcuser
```

Install the Requirements

It is best to first check that the server is up-to-date by running the command:

```
sudo apt-get update
```

You will also need to install Screen which will keep the Minecraft server software running if your connection is dropped:

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

Install the Minecraft Server

Start by creating a directory to store the Minecraft server and switch into it:

```
cd /
sudo mkdir minecraft
sudo chown mouser minecraft
cd minecraft
```

Inside that folder, download the PocketMine Minecraft server.

```
wget -q -O - https://get.pmmp.io | bash -s -
```

The installer will now install PocketMine

```
[*] Found PocketMine-MP Final_1.5dev (build 1254) using API 1.12.0
[*] This development build was released on Sat Jun 20 09:45:04 CEST 2015
[*] Installing/updating PocketMine-MP on directory ./
[1/3] Cleaning...
[2/3] Downloading PocketMine-MP Final_1.5dev-1254 phar... done!
[3/3] Obtaining PHP: detecting if build is available...
[3/3] MacOS 64-bit PHP build available, downloading
PHP_5.6.10_x86-64_MacOS.tar.gz... checking... regenerating php.ini... done
[*] Everything done! Run ./start.sh to start PocketMine-MP
```

Since we are using screen, you can start the server by running:

```
screen -S "Minecraft Server"
./start.sh
```

You will now be asked to select a language by entering the 3 character code:

```
[?] Language (eng):
```

Next you will be asked to accept the license by typing Y and pressing enter

```
[?] Do you accept the License? (y/N):
```

Now you will be asked if you want to skip the setup wizard. Type N and press enter and answer the questions

```
[?] Do you want to skip the set-up wizard? (y/N): N
[*] You are going to set up your server name.
[*] If you don't want to change the default value, just press Enter.
[*] You can edit them later on the server.properties file.
[?] Give a name to your server (Pocket-Mine-MP Server): State-Library-MC
Server
[*] Do not change the default port value if this is your first server.
[?] Server port (19132): 19132
[*] Choose between Creative (1) or Survival (0)
```

```
[?] Default Game mode (0): 1
[?] Max. online players (20): 20
[*] The spawn protection disallows placing/breaking blocks in the spawn zone
except for OPs
[?] Enable spawn protection? (y/N): y
[*] An OP is the player admin of the server. OPs can run more commands than
normal players
[?] OP player name (example your game name): SLQ-GM
[*] The white-list only allows players in it to join.
[?] Do you want to enable the white-list? (y/N): y
[!] You will have to add the players to the white-list
[!] Query is a protocol used by different tools to get information of your
server and players logged in.
[!] If you disable it, you won't be able to use server lists.
[?] Do you want to disable Query? (y/N): n
[*] RCON is a protocol to remote connect with the server console using a
password.
[?] Do you want to enable RCON? (y/N): n
[*] Getting your external IP and internal IP
[!] Your external IP is X.X.X.X. You may have to port-forward to your
internal IP X.X.X.X
[!] Be sure to check it, if you have to forward and you skip that, no
external players will be able to join. [Press Enter]
```

The server is now started and you connect a Minecraft client to the server on port 19132.

Commands

While the Minecraft server is running, you can issue commands within the server console. The main commands you can use are:

stop - This stops the server and saves the worlds to disk
gamemode creative <player> - Changes the specified player to CREATIVE
time set 6000 - Sets the worlds time back to midday

To suspend the Screen instance press **ctrl-a d**. This will return you to the main console. The previous Screen instance will still be running in the background. To return to that instance, type:

```
screen -R
```

To close the Screen instance, run the command:

```
exit
```

The Minecraft server is still running. You will need to run **stop** to stop the server first.

To find out the IP address of the server, run the command

ifconfig

again, you cannot do this while the Minecraft server is running. You will need to either stop the server, or suspend the Screen instance and return to the Screen instance later.

Launch Server as a Service

Next we can make a system service to launch our server. This means we can make it launch on start-up, and control it using ubuntu's built in service called systemd.

First we will make a service - called minecraftserver.service. Create it using the nano editor with the command:

```
sudo nano minecraftserver.service
```

Then paste the code below:

```
[Service]
WorkingDirectory=/home/mcuser/minecraft
User=mcuser
#Group=minecraft
Type=forking
# Run it as a non-root user in a specific directory
ExecStart=/usr/bin/screen -h 1024 -dmS minecraft ./minecraft_server.sh
# I like to keep my commandline to launch it in a separate file
# because sometimes I want to change it or launch it manually
# If it's in the WorkingDirectory, then we can use a relative path

# Send "stop" to the Minecraft server console
ExecStop=/usr/bin/screen -p 0 -S minecraft -X eval 'stuff \"stop\\\"\\015'
# Wait for the PID to die - otherwise it's killed after this command
finishes!
ExecStop=/bin/bash -c "while ps -p $MAINPID > /dev/null; do /bin/sleep 1;
done"
# Note that absolute paths for all executables are required!

[Install]
WantedBy=multi-user.target
```

This service will start up screen (as mcuser) then run a script in the working directory called minecraft_server.sh. To turn this into a service, we need to make it executable with

```
sudo chmod 775 minecraftserver.service
```

Then add it to our systemd directory with

```
sudo cp minecraftserver.service /lib/systemd/system
```

Server Start-up Script

Before we test our service, lets make our minecraft_server.sh script with nano

```
nano minecraft_server.sh
```

and enter

```
#!/bin/bash
./start.sh
```

Now this script needs to be executable

```
chmod 775 minecraft_server.sh
```

Test the Service

Nearly there! Lets start our service

```
sudo systemctl start minecraftserver
```

and check its running with

```
screen -r
```

All good? Now exit screen with ctrl+a+d

Then enable on start-up with

```
sudo systemctl enable minecraftserver
```

Its time to stop the service and reboot your server to check it starts properly:

```
sudo systemctl stop minecraftserver
sudo reboot
```

Whitelist

If your server is open to the outside world, we highly recommend that your server have a user whitelist enabled. This gives you to control who can join the server, allowing you to block users that

should not be connecting to your activities.

To edit your whitelist, enter the following command in the terminal:

```
nano white-list.txt
```

This will open the white-list file. Each user you would like to be able to connect to the server must be entered here, 1 username per line. Once you have finished, press **ctrl-o**, then **ctrl-x**.

Other Information

Your server will require internet access in order to authenticate the Minecraft clients.