



# **create\_server**

**SLQ Wiki Fabrication Lab 2024/04/25 23:12**

# Creating a Minecraft Server

This tutorial is for creating a Minecraft server using the CraftBukkit Mod on Ubuntu 18.04 with 2GB or RAM. 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 logged in as root user 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)

```
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
```

Now we need to make sure that Java is installed on your server. You can check this by typing the command:

```
java -version
```

If Java is not installed, you will get a message saying "Command 'java' not found". You can then download java through apt-get (don't use the command it suggests)

```
sudo apt-get install default-jdk
```

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:

```
mkdir minecraft  
cd minecraft
```

Inside that folder, download the CraftBukkit Minecraft server. For this example, we are using version 1.12.2. If you want to use a earlier/later version, visit <https://getbukkit.org/download/craftbukkit> and use the download link provided.

```
wget -O craftbukkit-1.12.2.jar  
https://cdn.getbukkit.org/craftbukkit/craftbukkit-1.12.2.jar
```

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

```
screen -S "Minecraft Server"  
java -Xmx1024M -Xms1024M -jar craftbukkit-1.12.2.jar nogui
```

The launching text should look something like this:

```
Loading libraries, please wait...  
[09:30.16 INFO]: Starting minecraft server version 1.12.2  
[09:30.16 INFO]: Loading properties  
[09:30.16 WARN]: server.properties does not exist  
[09:30.16 INFO]: Generating new properties file  
[09:30.16 WARN]: Failed to load eula.txt  
[09:30.16 INFO]: You need to agree to the EULA in order to run the server.  
Go to eula.txt for more info.  
[09:30.16 INFO]: Stopping server
```

Now you need to agree to the EULA. You can do this by running:

```
nano eula.txt
```

which will show you the contents of the eula.txt file. Using the cursor keys, change the line

```
eula=false
```

to

```
eula=true
```

then press `ctrl-o` to write the file, press `enter` to overwrite the eula.txt file, then press `ctrl.x` to exit nano.

Now run the java command again:

```
java -Xmx1024M -Xms1024M -jar craftbukkit-1.12.2.jar nogui
```

and you should see the following output:

```
Loading libraries, please wait...
[09:43.41 INF0]: Starting minecraft server version 1.12.2
[09:43.41 INF0]: Loading properties
[09:43.41 INF0]: Default game type: SURVIVAL
[09:43.41 INF0]: Generating keypair
[09:43.42 INF0]: Starting Minecraft server on *:25565
[09:43.42 INF0]: Using epoll channel type
[09:43.43 INF0]: This server is running CraftBukkit version get-Bukkit-
e60fc34 (MC: 1.12.2) (Implementing API version 1.12.2-R0.1-SNAPSHOT)
[09:43.43 INF0]: Preparing level "world"
[09:43.45 INF0]: Preparing start region for level 0 (Seed:
-1369994844442303813)
[09:43.46 INF0]: Preparing spawn area: 3%
[09:43.47 INF0]: Preparing spawn area: 31%
[09:43.48 INF0]: Preparing spawn area: 64%
[09:43.49 INF0]: Preparing spawn area: 92%
[09:43.50 INF0]: Preparing start region for level 1 (Seed:
-1369994844442303813)
[09:43.51 INF0]: Preparing spawn area: 5%
[09:43.52 INF0]: Preparing spawn area: 29%
[09:43.53 INF0]: Preparing spawn area: 57%
[09:43.54 INF0]: Preparing spawn area: 83%
[09:43.55 INF0]: Preparing start region for level 2 (Seed:
-1369994844442303813)
[09:43.56 INF0]: Preparing spawn area: 7%
[09:43.57 INF0]: Preparing spawn area: 49%
[09:43.58 INF0]: Preparing spawn area: 79%
[09:43.59 INF0]: Done (12.606s)! For help, type "help" or "?"
>
```

You should now be able to connect a Minecraft client to the server on port 25565.

## 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
```

Note that if you receive the result

```
[10:19:48 INFO]: Unknown command. Type "help" for help.
```

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:

```
nano minecraftserver.service
```

Then paste the code below:

```
[Service]
WorkingDirectory=/home/mcuser/minecraft
User=mcuser
#Group=mincraft
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
java -Xmx1024M -Xms1024M -jar craftbukkit-1.12.2.jar nogui
echo hi
```

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
```

## Other Information

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