Raspberry Pi Minecraft Server

A Raspberry Pi Minecraft server is a cost-effective way to have your very own private server with your friends or just for stuffing around on.

I have mine running for quite some while now and have not had to face any problems. It is important that you know that this server is not capable of hosting many people.

From my experience, anything over five people and the server will start to get a bit wonky. The best number of people I have found is between two and three. But, you can tweak the server to make it more efficient.

We will also be using spigot Minecraft because the default install didn’t work for me. Let's talk about games I found that it will crash a lot, be laggy, and is very unstable. The official Java version may improve in the future.

We will be using Java that should already exist on the Raspberry Pi (if Raspbian is the full version).

If you have a Raspberry Pi 2, 3, 4, or 4, we will make some changes to optimize it. We will also set it up so you can access it on the web and reboot if the Pi goes offline for any reason.

Equipment

Below is the equipment that I used to create this Raspberry Pi Minecraft server.

Recommended

Raspberry Pi MicroSD Card Ethernet Cable or WiFi (I recommend using an Ethernet for the best network performance).

Power Supply

Optional

USB Keyboard

USB Mouse

HDMI Cable

Raspberry Pi Case

Video Tutorial

Check out my video below if you want to learn how to set it up. A detailed text explanation is available below the video.

If you enjoyed the video, please subscribe or follow us socially to keep up with what's happening.

Setup the Raspberry Pi Minecraft Server

First, we'll need to install Raspbian into the Raspberry Pi. If you haven’t done so already, check out my amazing guide to installing NOOBs Raspberry Pi. It will walk you through all of the steps you'll need.

Important: Raspbian Buster will require you to have the most recent version installed. If Java isn't installed, then this will cause the server not to work. I recommend that you update to Jessie or Stretch, or Buster.

This tutorial will be performed entirely in the terminal. It is probably best to just boot into the terminal and not load the GUI.

1. First, let's bring Raspbian up to the latest version by entering the following.

2. We will now need a few modifications to the config. Let's start the tool by entering the following line.

Our guide contains more information about the raspi-config tool.

3. First, go to Advanced Options->Memory Split. Update this to 16. This will allow the server to have more memory.

4. Also, you don’t want to boot into Raspbian desktop. So make sure that the boot option is set instead to the CLI (Command Line Interface). This will give your server the best possible processing power.

5. If possible, change the overclocking setting to High

6. Finally, if you don't already have SSH enabled, enable it to remotely access the Pi.

7. Now, restart and finish.

8. Now we need the IP address of our Pi to connect to our server. To get the Raspberry Pi IP address, enter the hostname command.

You might set up a static IP address to ensure that your IP address doesn't change.

8. Next, we must ensure that Java and Git are installed. Without Java and Git, we won't be capable of building or launching the server.

Enter the following command for Raspbian's default JDK package and Git software installation.

9. Now we need the Minecraft server files. We'll use a Spigot builter tool to do this.

10. We will now need to run build tools. It creates Spigot server. It will take approximately 15-30 minutes to complete.

Add --rev 1.14.4 to the end of the command to get the latest version. Change 1.14.4 in your command to the most recent version number.

Important: If you have a Raspberry Pi A+, B, or any other variation of the Raspberry Pi 2, then the build tool will likely fail. You will need to instead generate the spigot.jar on a more powerful computer.

11. To check if the Spigot server has been successfully downloaded, use ls.

Keep your files in the /home/pi/minecraft directory as we need them to create all the server files. You can't start the server from a different folder. It will create the files there.

12. Now, we are ready to launch our server. To do so, enter this command. (Depending on which version you're using, you might need to change the version number. spigot-1.14.4.jar)

Raspberry Pi 1

Raspberry Pi 2, 3, or 4

You can increase your Xmx value by purchasing the Raspberry Pi 4 in 2GB or 4GB variants.

The server will automatically stop because we must agree to the Eula. This can be done by entering the following command into the Eula.

13. Here, change false to TRUE. Once done, save and exit by pressing CTRL+X then Y.

14. Now relaunch the server, it will take a while to create a map so give it about three to five minutes. If the map was already created, it'll only take thirty seconds to load.

15. The server should now be available on the local network and running.

16. You might want to mod your user now so that you can use all the server commands when you log in. Accessing the server backend will be slightly more difficult if it is set to auto-boot at startup.

To mod your user, simply run the following command when the server has launched (Replacing username with your username).

17. The Minecraft server on the Raspberry Pi will now be up and running fine, but you may want to do some optimizations to the server to make it run even better.

Connecting to Minecraft Server

If you're connected to your local network, it should not be difficult to connect to the Minecraft server that runs on the Raspberry pi. To test it out, do the following steps.

Load the Minecraft Java clients on a computer in the same network as the Pi.

You might see your server in the local listing if you go multiplayer. If it doesn’t, go to direct connect and enter your IP address using the command hostname-I.

Port forwarding is necessary if you wish to allow access to Minecraft servers via the internet.

If you are interested in how to do it, then check out my guide on Raspberry Pi Port Forwarding. You will need to port forward the port 25565 (unless you change it in the server properties) to the IP of your Pi.

Here are some tips to configure the server and get it up and running.

Optimizing the Minecraft Server

We will now install a plugin to optimize the performance of the Raspberry Pi's Minecraft server.

Let's first install NoSpawnChunks, which will help to stop the Minecraft server from eating up too much RAM.

There are many plugins that can increase performance and extend the server functionality. You can use the command wget to download them to Pi like we did.

Editing the Minecraft Properties

You probably want to learn how to modify the server properties. This ability is vital for optimizing the server, customizing it to your liking, and ensuring that it runs smoothly.

Here are more details about each of the server settings.

Enter the following line to gain access to the server properties.

To optimize the server's performance, we will need to modify a few settings.

You can change these and other settings however you like, but keep in your mind that the Pi can't process too much.

Startup Bootcamp

To have the server start on boot, we will need to do a few extra steps.

1. We will need to create a service for the Minecraft server so let's start writing the service file by entering the command below.

2. The following text will be required to be entered into this file.

This file defines the service, so the service manager knows how and what to run. Make sure to update the version number of your spigot whenever you upgrade.

Once done, save the file by pressing CTRL + X then Y followed by ENTER.

3. Now, we will need to enable the service. You can enable this service by running the following command.

4. You should now be able to start the Minecraft server by simply using the following command.

5. Using a similar command, you can check on the status of the service. Checking the status is great for debugging.

5. You can stop the server by using the following command.

Your server should now start on boot. It is possible to test it by restarting your Raspberry Pi. It will take a few minutes to startup.

If you want to get access to the server on the command line, then you will need to shutdown the server and load it using the normal command.

I hope you found this tutorial helpful in setting up a stable Raspberry Pi Minecraft server. If you liked this tutorial then please visit our other Raspberry Pi Projects.

Also, feel free to drop us a comment below if you have better optimization settings, plugins, or ideas. Let us know if you have any problems.