

# How To Create A Minecraft Server For The Raspberry Pi 4 With Balena

This project and guide is a community contribution by Alex K., aka AlexProgrammerDE. Check out his GitHub page, his GitHub repository, and give a Star. Enjoy the guide.

If you want to host your own Minecraft Server -- all from a Raspberry Pi -- this project is for you! Try the balena Minecraft Server, a starter project to quickly and easily create a Minecraft Server and use SCP and RCON to manage it!

Minecraft is a cool adventure game with multiplayer functionality. There are many Minecraft server hosting companies around the world, that charge varying prices, but with this project you can host a Minecraft server yourself for free! With balena Minecraft Server, you can host and manage your own server right on a Raspberry Pi 4. This blog post walks you through the entire process, and shows you how to build and connect to your own Minecraft server quickly and easily by using balenaCloud.

All you need is a Raspberry Pi 4 and a little bit of time. We've done all the hard work by configuring the Server & the other services, and setting everything up in a repeatable format to get you up and running with minimal effort. It's a great introduction if you've never tried a project like this before. Let's get to it! Hardware required

- Raspberry Pi 4B (We recommend the 4GB model. 1GB is not enough!) - A fan or cooling system to prevent lag caused by throttling - A 16GB or greater micro SD Card (we always recommend SanDisk Extreme Pro SD cards) - Power supply

## Software required

- A balena Minecraft Server download from GitHub – Software for flashing an SD card (we suggest balenaEtcher). – Free balenaCloud account to set up and manage your Pi. – Download the balena CLI toolkit - to be installed on you computer. This will allow the Pi to install the project-code.

## Setup the Raspberry Pi

Once you've found all the hardware and prepared all the software, we're going to start setting up the Raspberry Pi. Sign up for a free balenaCloud account

The first thing you'll need to do is sign up for an account if you haven't done so already. If

you've already got a GitHub or Google account, you can use one of those accounts as a single sign on method. Create a balenaCloud application

Follow the instructions on the user interface to add an application, selecting the correct device type for the device you're using. The easiest way would be to choose Starter as the application type, then hit Create New Application. Using the starter application provides you with all of the features of the microservices application and is free up to and including your tenth device. Once you add the application, you'll arrive at the dashboard for your newly created application. If you choose to, you can rename your application. Note: You'll need to remember that name for later when you push your code. Add a device and download the balenaOS disk image from the dashboard

Add a device within that application by clicking the 'Add Device' button. When you add a device you specify your device type, which is important that it matches the device you're using. If you are connecting to a wireless network, you can set your WiFi SSID and passphrase here too. Otherwise, a wired connection will suffice.

This process creates a customized image configured for your application and device type, and includes your network settings if you specified them. Note: When you're first getting started, a development image will be most useful, as it permits a number of testing and troubleshooting features. More details on the differences between development and production images can be found [here](#). If you're confident you can go ahead and deploy the production image straight away. Flash your SD card with the balenaOS disk image and boot the device

Once the OS image has been downloaded, it's time to flash your SD card. You can use balenaEtcher for this.

Once the flashing process has completed, insert your SD card into the Raspberry Pi and connect the power supply.

When the device boots for the first time, it connects to the balenaCloud dashboard, after which you'll be able to see it listed as online and move onto the next step.

Troubleshooting: It should only take a few minutes for the new device to appear in your dashboard. If your device still hasn't shown up on your dashboard after a few minutes, something has gone wrong. There's an extensive troubleshooting guide in the documentation, with lots of information on why this could be, but if you still can't get your device online, come on over to the forums where we'll be able to help out. Deploy the project code

Now you've got your Raspberry Pi online within the balenaCloud dashboard, it's time to deploy the project code and transform your Pi into a Minecraft server! Download the project from GitHub

Grab a copy of the balena Minecraft Server project from GitHub. You can download the ZIP from GitHub as shown below, but if you're familiar with Git you can use git clone. Important note for Windows users: the combination of git clone and balena push can cause issues due to line ending changes. We recommend using a combination of either the zip download of the project and balena push or if you'd like to use git clone then also use the git deployment method git push instead of balena push. Push the project code to your Raspberry Pi

After installing the balena CLI and downloading the balena Minecraft Server software from GitHub, you can push the code. Before you can push don't forget to unzip the file you just downloaded. From within the unzipped file, execute balena push appName in a terminal, where appName is should be the the application name you set earlier in the guide. For example: balena push balenaMinecraftServer. If everything worked out correctly, after a few minutes your device information screen in the dashboard should look something like this, showing the service running. At this point you're ready to move on, connect things up and give it a try! Give it a try

Once you've got your Pi powered up and your server booted, you're ready to go! Connect to the server via Minecraft (Java Edition). Note: This testing example only works if your Raspberry Pi 4 and your computer are in the same network. We have instructions on creating worldwide multiplayer mode later in this article.

You can now enjoy playing Minecraft survival together with your friends! What next?

Servers don't run and maintain themselves! Here are a few tips on accessing your server remotely via RCON to configure your game and how to edit all server files via SCP. Minecraft servers Connect to the terminal

Patching and administering changes to your server requires connecting to your terminal via RCON. The port is 25575 and the password is balena. It allows you also to op yourself and to run other commands. Here are some preferred RCON clients: mcrcn NOTE: You will need this batch file if you are using Windows. Just paste this in the unzipped directory. Minecraft Server RCON

Edit files

You can connect to the server and change your serverfiles. I recommend using a tool like WinSCP, or if you are using OSX or a linux distribution, you can use Filezilla. The IP Address

to connect to is "balenaminecraftserver" (without the quotes), the protocol to choose is SCP (if you have the choice), the port number is 22, the username is "root" (again, without the quotes), and the password is "balenaserver" (no quotes). The files are in the folder named "serverfiles" at the root directory. You can double click to open that directory and browse the files in there. NOTE: You can also change your SCP password by setting the SCP\_PASSWORD Environment Variable within balenaCloud. On the left menu, click on Device Variables, and then click the Add Variable button. Give it a name of SCP\_PASSWORD, and set the value to your password. The result should look like this:

## Connect to another Wifi

balenaMinecraftServer has wifi-connect integrated. This will allow you to take your Pi with you anywhere, and still make use of it! If you want to read more, here is a link about how it works. Double RAM

Devices like the Raspberry Pi 4B 4GB have enough RAM to run the server with 2GB RAM (the default value used by a Minecraft server is 1GB). If you set DOUBLE\_RAM to true it will double the amount of RAM used by the server. Change hostname

You can change the hostname by defining the DEVICE\_HOSTNAME Environment Variable within balenaCloud. NOTE: If you decide to change the hostname, you will have to use your new hostname within Minecraft to connect to the server, instead of balenaminecraftserver. Add plugins

Nothing keeps a game more interesting than its plugins. You can add plugins onto your balena Minecraft Server by adding preferred plugins into the plugins folder using SCP (The folder is here: /serverfiles/plugins/). The current Minecraft version is 1.15. You can get your plugins from here (other sites are available too. ): Spigot Bukkit

NOTE: Before adding the plugin, check to ensure that it supports Minecraft version 1.15. Play worldwide

Once you've perfected the setup of your server on your local network, you might be interested in unveiling your server to the rest of the world! Here's how you can enable remote access and allow players to connect via the Internet. Setting up Dynamic DNS

If you'd like to allow friends outside of your local network to join your server, you'll need to set up dynamic DNS (DDNS) to expose your Pi to the outside world.

This example uses a service called No-IP, which has a free tier for people who want to try DDNS out, though other options and methods do exist as well. In the case of this example, you will need to: - Create an account with No-IP by visiting their website. - After creating the

account and logging in, create a Hostname (example: balena.serverminecraft.net) by following their documentation. - Set up Port Forwarding: You will need to route your Minecraft traffic to port 25565 on your Pi. To do this, you will log in to your home router and setup Port Forwarding. This step varies by particular brand of modem or router, but the No-IP documentation does a good job of describing the process here. You may need to follow instructions specific to your modem or router if the No-IP documentation does not contain your particular type. - Optional: You can login to No-IP with your router to keep the IP Address current in case it changes. That allows the router to connect automatically to No-IP. Here is a guide by No-IP on how to accomplish this. - Paste your public / external internet address in the box labeled IP Address into the No-IP dashboard. You're done.

For a deeper look at setting up remote access, please reference this guide (Note: You can skip the DUC part). Custom Server

If you want to customize your server even further, but don't know where to start, take a look at some of the servers listed here for ideas: Spigot (Vanilla Java Edition) Craftbukkit (Vanilla Java Edition) Vanilla from Minecraft (Vanilla Java Edition) Paper (Vanilla Java Edition) Forge (Modded Java Edition)

Note: Balena Minecraft Server uses Paper. It is an efficient and powerful server. It is compatible with spigot and bukkit plugins. I encourage you to take your server build even further! There are many tutorials out there on server customization-- this article only touches on a few ideas. If you need help, please reach out by submitting an issue on GitHub. Until next time...

Thanks for taking the time to check out my guide, we hope you had success with the project and created a very cool Minecraft Server. If you have any trouble getting the project running or have any other feedback, we'd love to hear it; everything helps to improve our projects and tutorials for next time! Find, try, and submit your own projects on balenaHub

You'll find this edge project and many others at balenaHub, our marketplace for IoT and edge projects. With a quick search, find your next edge project, from digital signage, surveillance, edge AI, audio analysis, and more. Or create your own project and share it with the world. You can always find balena on their forums, on Twitter, on Instagram, or on Facebook.

Special note

This project and guide were written and contributed by balena Community Member Alex K., aka AlexProgrammerDE, who came up with the idea, built the containers required, and wrote up the material for this project. We're very grateful for his contribution! Be sure to check out his GitHub page. Also, be sure to check out his GitHub repository, and give a Star!