

# How To Make A Minecraft Server (Java Edition) On Windows Or Ubuntu

Let's create a Minecraft server up and running in minutes for you to enjoy with close buddies.

Minecraft is a creator's paradise or a strange-looking blockified world annoying you to the core each time you step in.

Either way, gamers around the globe find it irresistible and play Minecraft in big numbers, making it the best-selling video game of all time.

While you can play Minecraft solo, the experience goes bland after you end up spending hours inside mining and crafting.

Ergo, you have the multiplayer to keep the fun alive. Now you can either join the big-name servers and play along with hundreds if not thousands of players.

Or you can make a Minecraft server yourself to have fun with your dear ones.

It's important to know this guide will create a Minecraft server on your personal PC. So you shouldn't host players you don't trust, as you'll reveal your IP address.

And please remember, we'll be hosting Minecraft Java edition, which works on Windows, macOS, and Linux.

Now that you know enough, let's start with the real deal.

Most likely, you would be using your own PC for this. However, you can also opt for Minecraft Server Hosting for additional support and professional gameplay for a large number of players.

But, we'll stick with the assumption that you're using a personal computer.

And since we are installing Minecraft Java edition, we need to first install Java or update if you already have it.

Installing Java on Windows is fairly simple. Visit this [Java download page](#), and grab the latest version.

For Ubuntu, open the terminal and enter `sudo apt install openjdk-18-jre`, and press enter. As of this writing, Java 18 was the latest; you should modify it accordingly.

If you are to run Minecraft on a 3rd-party cloud server like DigitalOcean, install java with this command instead: `sudo apt install openjdk-18-jre-headless`. This signifies you'll run Minecraft without a graphical user interface.

Subsequently, download and install the Minecraft Java Edition Server to get a single file named `server`. Just so you know, there aren't separate files for Windows and Linux.

After downloading, it will be ideal to create a separate folder for this since the server will generate more files that you would want in a nice tidy place instead of lying across the main download folder.

Next, you can choose to have a Minecraft server with or without a graphical user interface (GUI). While having a GUI is convenient, you can have some extra resources if you choose to go by the command prompt way.

Still, we'll see both.

### Minecraft Server with GUI

Using the Minecraft server with a GUI is almost identical in Windows and Ubuntu.

Run the server file.

This won't open anything, instead will create more files, including a text file for the end-user license agreement (`eula.txt`). Open this text file, replace `eula=false` with `eula=true`, and save the changes.

This indicates that you agree with the developer's terms and conditions. If you miss this step, the server will shut down once you open it.

Afterward, double-click the server file again to open its graphical user interface. The will start

loading all the prerequisites, and you will get a Done on the bottom right (just above the empty field), indicating the server is ready for hosting.

Open the terminal and use the `cd` command to navigate to the Minecraft folder. Alternatively, you can open the specific folder, right-click, and select Open in Terminal to have it directly point towards that directory.

Now run `java -Xmx1024M -Xms1024M -jar server.jar`, and press enter.

The `Xmx`, `Xms` symbols denote the maximum and minimum ram that Minecraft servers can run with. You can experiment with these values to get an optimal experience. For instance, I can change it to: `java -Xmx4096M -Xms2048M -jar server.jar`, which means the Minecraft server will operate between 2Gb to 4Gb RAM.

This is prompting to accept the EULA before running the server. Similar to what we did in Windows, locate the newly created `eula.txt` file, and modify it by changing `false` to `true`.

Afterward, paste and run the same command:

This will take some time, but you will soon see a graphical user interface pop-up, indicating the server status.

You may also encounter some errors like Failed to load module "canberra-gtk-module". This can be removed by installing the specific files by opening a separate terminal and running `sudo apt install libcanberra-gtk-module libcanberra-gtk3-module`.

Now restart the server, and you won't get this error again.

While having the GUI is great, using the terminal each time to open it can be annoying. So we'll make a script to run it easily.

For this, create any text file in the folder and name it as `start.sh`. Next, open this file with any text editor. Then, copy-paste the `java -Xmx1024M -Xms1024M -jar server.jar`, and save the changes.

Finally, right-click over it, enter Properties at the end of the pop up, select Permissions at the top, and check Allow executing file as program.

Now all you have to do is right-click the start.sh file, and select Run as a program to open the Minecraft Server GUI.

This was how to create a Minecraft server that has a GUI. Let's check out the alternative.

## Minecraft Server Without GUI

Arguably, this is a more powerful option of the two and makes more sense for the power users.

### Windows

In this method, we won't be running the server.jar file directly. Instead, we'll create one start.bat file to run the server in the command prompt.

First, open the folder containing the server.jar and create a start.txt file. Now open the file in Notepad and copy-paste the following code into the text file, and save the changes:

Next, rename the file from start.txt to start.bat.

Now, double-click this start.bat file to run the server the first time. This will create multiple files with the command prompt closing moments after running the bat file.

Proceeding further requires modifying the eula.txt file, similar to what we did in the GUI method.

Once you do that and rerun the bat file, this will be the output:

The 'Done' at the end signifies the server is running perfectly.

### Ubuntu

No GUI in Ubuntu means adding a nogui in the script file that we earlier made in the Ubuntu GUI method. So the new script will be: `java -Xmx1024M -Xms1024M -jar server.jar nogui`

Subsequently, save the changes and run the script to open the Minecraft server in the terminal:

You can type help to get the list of valid commands, or you can check the Minecraft Wiki.

Finally, we are done with the server creation part.

Now to play on this newly created server, all you have to do is locate the Add Server under Multiplayer in the Minecraft launcher. Up next, type localhost or 0 in the Server Address and click Done.

This will list the server in the multiplayer menu. Now select the server and click Join Server at the bottom.

This will open the game for the local player (you). But you haven't created this to play alone, so let's check how to play multiplayer on your native server.

## Hosting Players on Your Minecraft Server

You can host other Minecraft servers in your local area network (LAN) or elsewhere.

People on the same network just need the host's internal IP address, which you can get by typing ipconfig in the command prompt.

Ubuntu people can type hostname -I to get the internal IP address:

It's the starting of the result, 192.168.1.32, in this case.

Share this with other players, and they can join your server by listing this in the server address section inside the multiplayer tab.

However, Minecraft hosting for people outside the LAN will need some networking skills for:

Firewall configuration - Port forwarding

The first step is to configure the firewall to allow connections to a specific port number that you can find on the server properties file created inside the main server folder.

So the default port number is 25565.

Subsequently, you need to tell the firewall to allow connections via this port to let your Minecraft friends join in.

## Firewall Configuration

Firewall configuration depends on your operating system. Since this guide is meant for Windows and Ubuntu, we'll go through both.

### Configuring Firewall on Windows

Press +R, type `firewall.cpl`, and hit enter.

Basically, we'll be setting inbound & outbound rules for the port (25565) for UDP & TCP networking protocols. This will be a four-step process. Check this demo where we complete the first step by creating an inbound rule for TCP:

You will be repeating this for inbound (UDP) and outbound (TCP & UDP) for the same port.

### Configuring Firewall on Ubuntu

In Ubuntu, Firewall remains off by default, and you can verify that by running `sudo ufw status verbose` in the terminal.

If it returns with `Status: inactive`, you can activate it by `sudo ufw enable`.

Next, type `sudo ufw allow 25565`, and hit enter to open port 25565, which is the default to allow connections in Minecraft.

Now we can check the open port using `sudo ufw status verbose` again:

Now that we have this port ready for connections, it's time to make it available for others. This process is called port forwarding, which can take a few seconds to a minute. But we can't explain that here since that is based on your WiFi router.

Search for your device and be done with it.

Afterward, your friends will need your public IP address.

You can get this by just googling what's my ip address or visiting something like [whatismyipaddress](#).

Blurring my public IP address reiterates the fact that don't share yours with someone you don't trust.

The process to join this server is similar to joining your own server, the only difference being they will use public IP address:port number (ex. 125.245.74.894:25565) in the Server address section. And don't bother with the server name, it will automatically fetch that.

So that's how you make a Minecraft Java edition server and have your friends join it.

It's straightforward if the whole party is on the same LAN. let's talk about games This way, you just create a server and share your internal IP address.