

---

# *Setting Up a Minecraft Server*

---

## Introduction

So you've played Minecraft, but maybe you're bored with the constant zombie attacks, random creeper explosions, or (more likely) how lonely it can be playing Minecraft by yourself. Well, I have some good news! You don't have to play Minecraft by yourself! With a Minecraft Server, you'll be able to play, build, explore, and destroy with your friends!

In this document I'll explain some vocab you'll need to know as well as explaining the different kinds of Minecraft Servers (and setting them up). This document assumes that you have some knowledge of Minecraft itself as well as some basic computer networking knowledge. You'll need to know how to figure out the server's IP address, and how to open the proper ports to allow outside connections into your server. I'll explain what all that means later, but actual configuration won't be covered here.

**Note:** There are some very minor differences between running a Minecraft server on a Windows machine and running it on a Mac or Linux computer. I will primarily cover how to set up a Windows-based Minecraft server, but will point out the different steps when they occur.

By the end of all this, I hope to have given you all the information you'll need to create your own Minecraft Server from scratch.

## Vocabulary

**Minecraft Server** - This is a combination of a computer and some software that hosts a multiplayer Minecraft game. There are a few different types of server (as discussed next), and many MANY different ways to configure it to suit your needs.

**Vanilla Server** - This means that your server has \*no\* plugins. You're running the very basic version of a Minecraft Server.

**Bukkit Server** - These servers include plugins from the very popular plugins.bukkit.org site. **Bukkit** is also referred to as CraftBukkit and CB.

**Forge Server** – Basically Bukkit's fancier older brother. These servers feature expansive modifications and require the players to have the modifications installed on their computers as well. Setting these up is a bit of a pain, so they will be covered in another document.

**Throw-Away Server** - Also called a Temporary Server (or Temp Server). It's a one-use server for specialized events where you don't plan on hosting it for an extended period of time.

**Admin** - The police of your server. You should have 1-3 Administrators for a small server and more depending on how many people are usually on your server. These admin require OP powers (discussed next) in order to keep the peace.

**OP** - This is a set of powers bequeathed to those deemed reliable. It's essentially "God Powers" over the server. They can change the rules, create & destroy massive chunks of land, and kick & ban players. You should only OP your Admin.

**PvP** - Player vs. Player. You can configure your server to allow players to fight each other, or you can turn it off.

## Vanilla Servers

### Before You Skip Ahead:

Even if you would rather set up a Bukkit server, I strongly recommend reading through and following along with building and configuring a Vanilla server on your own computer. It is the easiest way to learn how to configure a server, and much of what you'll learn in this section will be referenced in the section on Bukkit servers.

That being said, let's get to it!

### Getting the necessary files:

- Hop on your web browser of choice and navigate to *minecraft.net/download*
- Under the heading for "Multiplayer Server", click the link for the .exe file if you're running Windows, or the .jar file if you're running Mac or Linux.

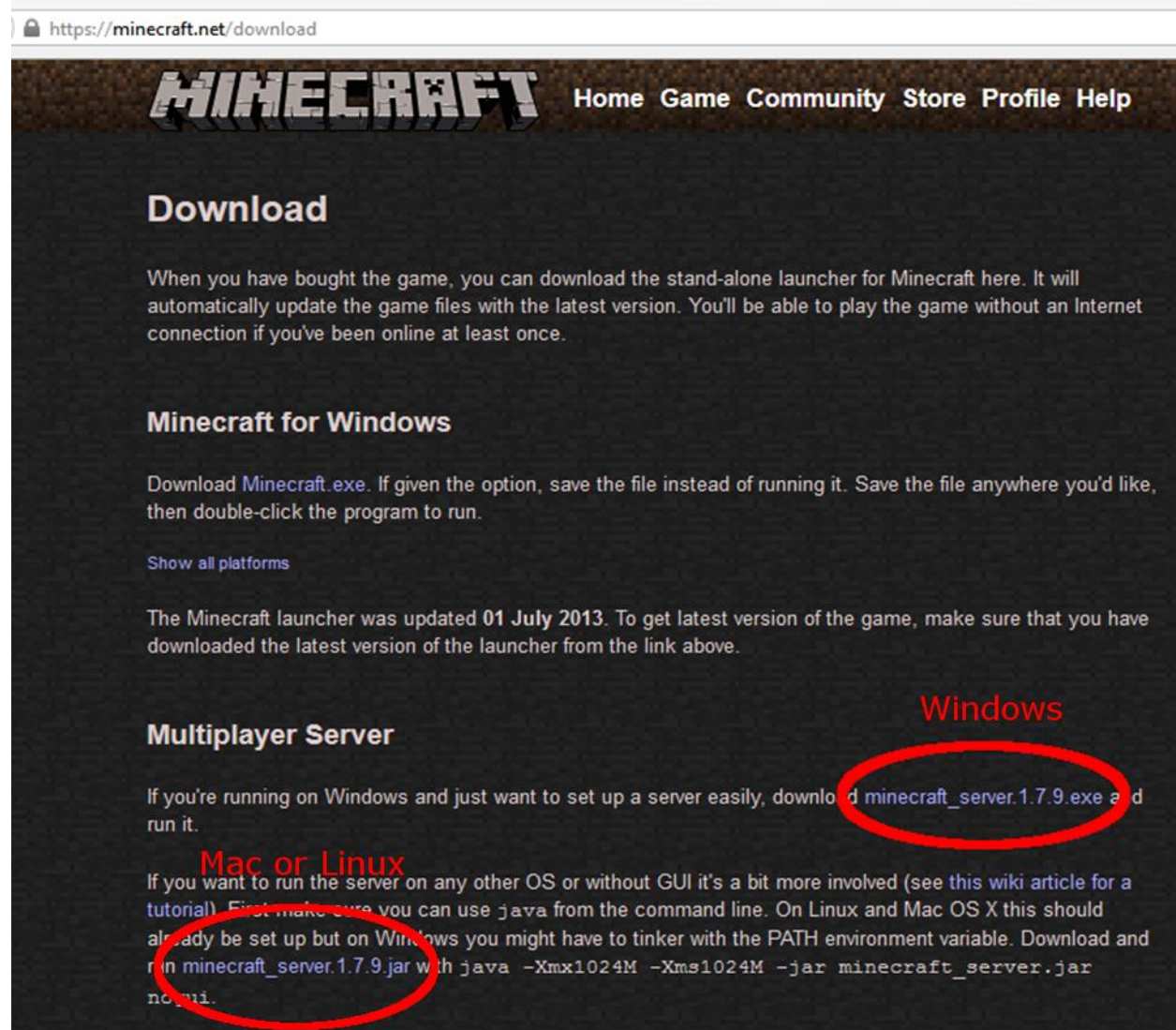


Fig. 1

- Save this file in its own folder, and run the .exe or .jar file you just downloaded.
  - **Mac:** You might have to right-click and tell it to run as a Java application. .jar files can be opened using a program like 7-Zip, and Macs tend to default to that.
  - **Linux:** You might need to give the .jar file permission to run as an executable. Right-click; under the “Permissions” tab there is an option for “Execute” or something similar.
- Once you run it, a white box should appear, and a whole bunch of new files will show up in the folder where you put the .exe/.jar file. It all should look something like Figure 2.

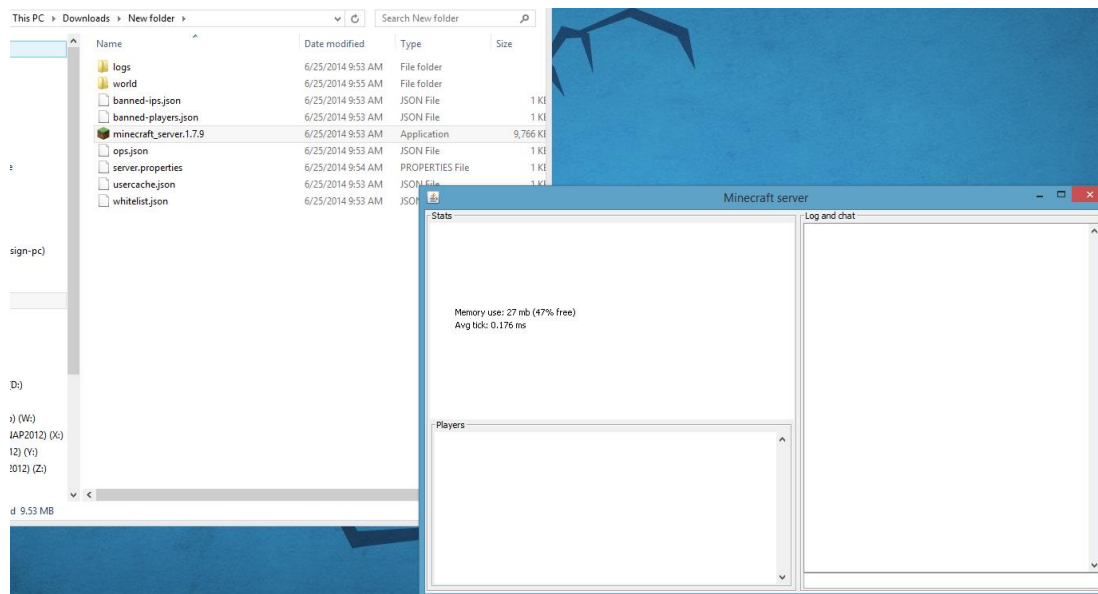


Fig. 2

- You’re now officially running a Minecraft server! Open up Minecraft, click on “Multiplayer”, then “Add Server”, and type in whatever you like for the server name, and the server address will be 127.0.0.1 (this is telling your computer to connect to a Minecraft server that it is hosting!).

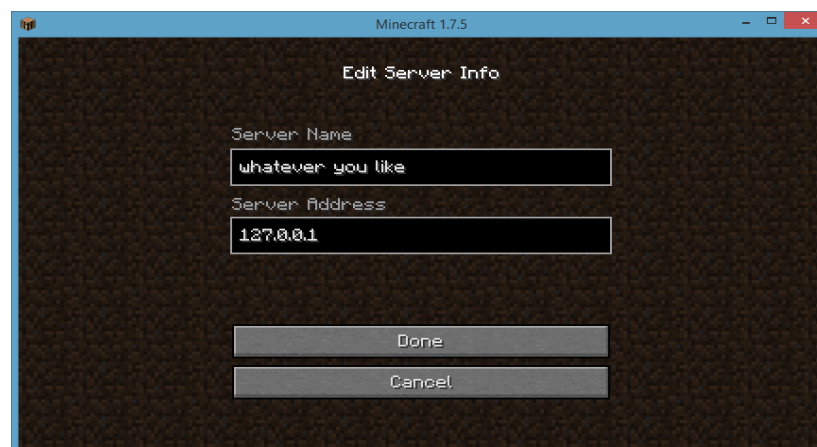


Fig. 3

- Click “Done”, click on the server you just made, then click “Join Server”

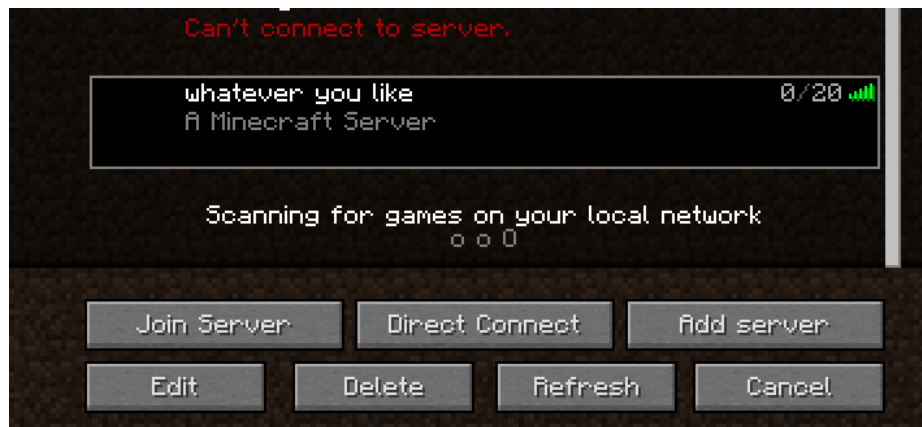


Fig. 4

- Woo! Minecraft! Of course, you’re the only one playing still, but we know it’s up and running! In order for other people to connect to your server, you’ll need to give out your IP address and make sure the proper ports are open on your network. I recommend consulting Google if you run into any specific issues.

## Configuration

That’s how you get the Vanilla server up and running! Pretty easy, yeah? Well, here comes the fun part: configuring it. Remember all those new files and folders that were created when you launched your server? Yeah, those are all very important. These are files and folders you will have to deal with no matter which type of server you want. Let’s take another good look at the main server folder.

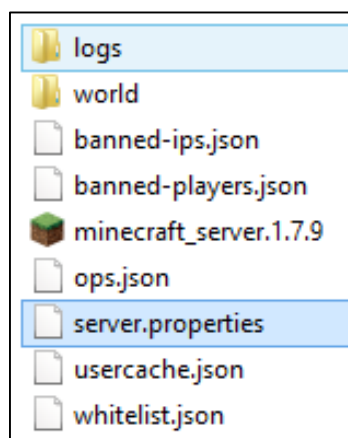
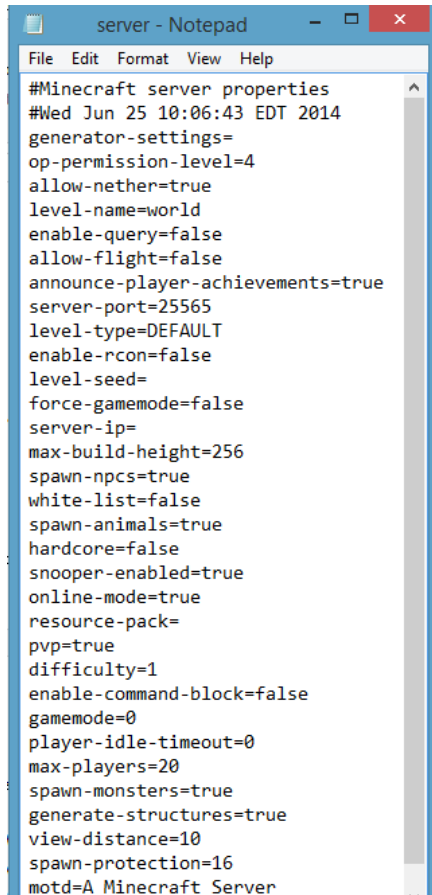


Fig. 5

Your folder may look slightly different depending on what type of server you’re running, and what version of Minecraft you’re running. However, these are the essential files that will definitely show up no matter what. Your main concern is the server.properties file.

## Server.properties

You can edit this file using Notepad or a similar plaintext-editing program. Do **not** edit this file in Word or LibreOffice unless you really know what you're doing. This is where you configure your server to suit your needs.



```
#Minecraft server properties
#Wed Jun 25 10:06:43 EDT 2014
generator-settings=
op-permission-level=4
allow-nether=true
level-name=world
enable-query=false
allow-flight=false
announce-player-achievements=true
server-port=25565
level-type=DEFAULT
enable-rcon=false
level-seed=
force-gamemode=false
server-ip=
max-build-height=256
spawn-npcs=true
white-list=false
spawn-animals=true
hardcore=false
snooper-enabled=true
online-mode=true
resource-pack=
pvp=true
difficulty=1
enable-command-block=false
gamemode=0
player-idle-timeout=0
max-players=20
spawn-monsters=true
generate-structures=true
view-distance=10
spawn-protection=16
motd=A Minecraft Server
```

Fig. 6

A lot of these entries should be self-explanatory; “max-players” is the maximum number of players allowed on the server. “spawn-animals” is whether the server should spawn cows, sheep, chickens, etc.

Check out [The Minecraft Wiki](#)'s page for the server.properties file for a detailed description of every single entry in this file.

The entries that you should note are “server-port”, “enable-command-block”, and “pvp”:

- **server-port:** Change this if you plan on hosting multiple Minecraft servers on one computer, or if you're getting port errors when launching the server.
- **enable-command-block:** Command blocks are VERY powerful tools for OP'd players on your server. Enabling them allow for OPs to use them.
- **pvp:** Allows/Disallows Player vs. Player fighting on your server.

**Note:** You will need to restart the server before any of these changes take effect. Moreover: if you tweak any of the world generation lines (i.e. ‘level-seed’, ‘generator-settings’, etc.) those changes will only take effect on parts of the world that haven't been explored yet OR if you delete the “world” folder and generate an entirely new world.

## The Rest of the Server's Folder

Most of the other files that show up after you launch your server for the first time are files that the server uses to store usernames and IPs.

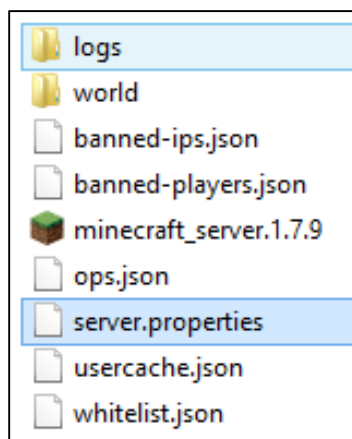


Fig. 7

The .json files in Figure 7 are there to explicitly control who can get on your server. If some undesirable person gets your server's IP address, you can ban that person's Minecraft username, or even the IP address that they logged in from using the OP commands (which I will discuss in a bit). You can even set up a whitelist to only allow certain people onto your server (check out the server.properties file to enable it).

The two folders generated are:

**Logs:** Your server keeps track of commands and login / logout information while it's running. All of this information is stored as a text file in this folder.

**World:** This folder stores your entire Minecraft world.

That's about it for configuration on a Vanilla Minecraft Server! The files and folders I've discussed are universal no matter what kind of server you're running, but with the addition of Bukkit plugins or Forge mods you'll have more configuration to do and more documentation to peruse. Let's get into that!



## Bukkit Servers

As stated earlier (and in other documents I've written), a Bukkit server uses special Bukkit "plugins" to add functionality to your server. They are easy to install and uninstall, but you will have to peruse the specific plugin documentation in order to learn how to use it. I strongly recommend that you do your research BEFORE implementing a plugin. It's no fun to be forced to regenerate a whole new Minecraft world due to a player exploiting a plugin's functions. I speak from experience.

There is also much more set up for a Bukkit server as opposed to a Vanilla server. I will do my best to keep things in simple English. There is a step-by-step guide found at [wiki.bukkit.org/Setting\\_up\\_a\\_server](http://wiki.bukkit.org/Setting_up_a_server), but I feel that some of the language and lack of pictures can drive away beginners. That being said: I am using that guide as a foundation for what I'm about to write.

## Getting the Files

A nice thing about Bukkit is that Windows, Mac, and Linux will be downloading the exact same file and most of the steps are the same. You will need the current release of Bukkit's .jar file. Head over to the wiki page I linked to in the previous paragraph, and down the page a little will be links for "CraftBukkit - Recommended Build". It should look something like Figure 8.



Fig. 8

Just like with the Vanilla server, you'll need to put this .jar file into its own folder somewhere on your computer since it will generate a bunch of its own files and folders when it is run for the first time.

## Writing the Launch Script

Here comes the fun part! We need to write a short script in order for this to run properly. Don't fret! Just follow the directions and everything will be just fine (AND you'll be able to say you've written a script!).

Open up Notepad / TextEdit / gedit / whatever plaintext editor you like and type the following:

```
java -Xmx1024M -jar craftbukkit.jar -o true  
PAUSE
```

What this does is tells the computer to run `java` using a maximum of 1024MB (you can tell it 1G or 2G as well) of RAM on the jar file `craftbukkit.jar`.

This script is all kinds of important for your Bukkit server! You can change the max amount of RAM your server is running by changing the `-Xmx1024M` entry. For example, if you want to run your Bukkit server with a max of 2GB of RAM, you can change it to `-Xmx2048M` or `-Xmx2G`. **Also note:** You will need to either change the name of the `.jar` file you downloaded, or the `.jar` file name in this script. As an example, the `.jar` file I downloaded just now is named “`craftbukkit-1.6.4-R2.0.jar`”. In order for this script to run properly, I will either need to change this file’s name to “`craftbukkit.jar`” (to match the script) OR change the script to read `-jar craftbukkit-1.6.4-R2.0.jar` (to match the file).

The next step is to save it as a `.bat` file instead of a `.txt` file. You can do this by going to “File” → “Save As...” and changing the “Save as type:” box to read “All Files” and naming the file ‘`startserver.bat`’ as shown in Figure 9.

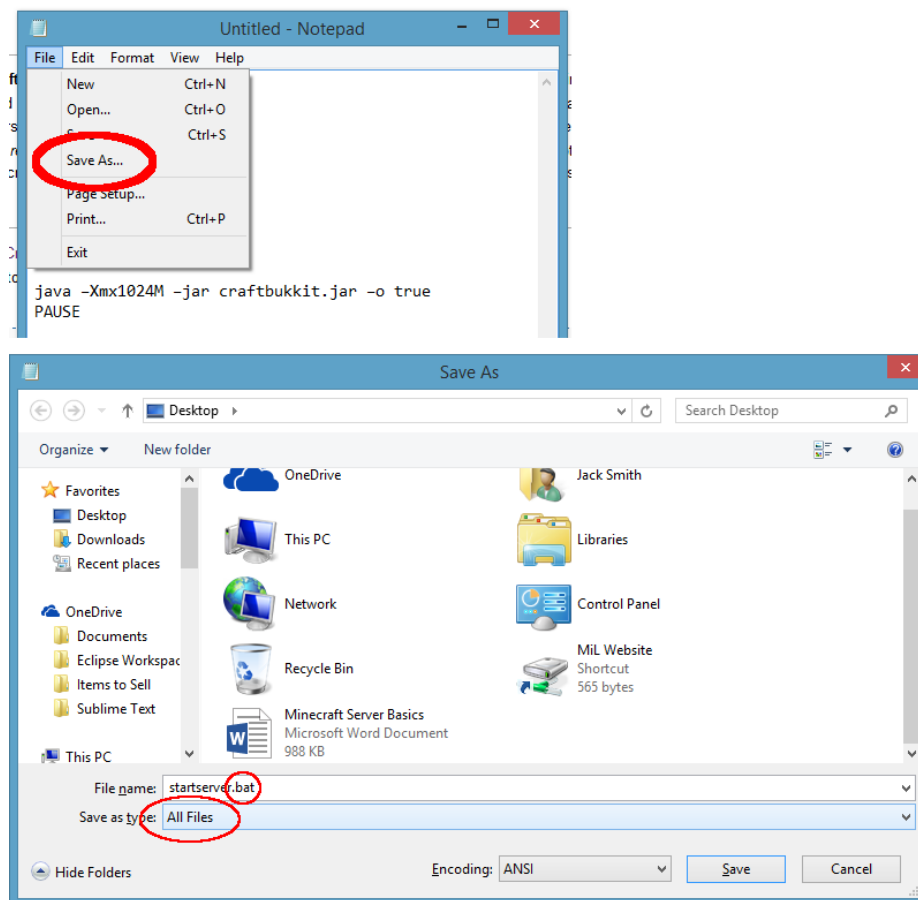
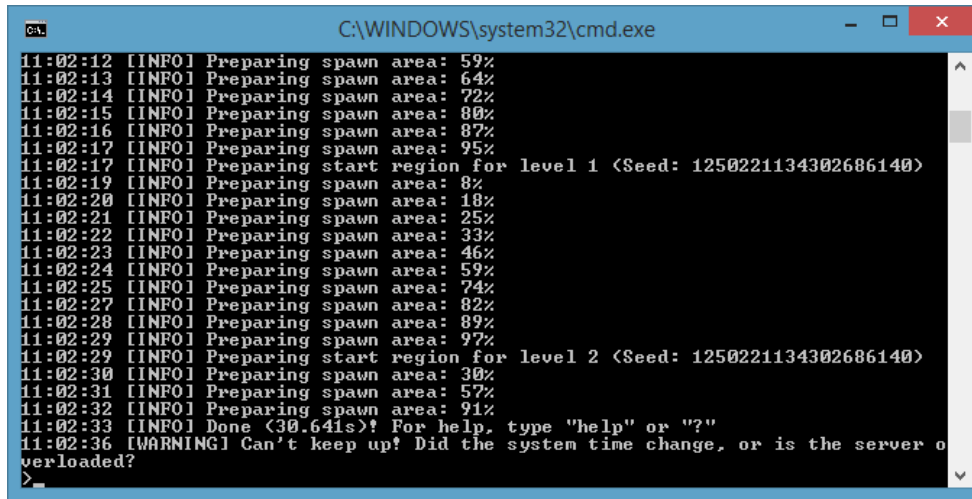


Fig. 9

Please make sure the script is saved as a `.bat` file on a Windows system. It is a very common mistake to have it save as a `.txt` file named `[insertname].bat.txt`. If you're having issues getting your plaintext editor to save as a `.bat` file, you might want to check out Notepad++ (Windows) or Sublime Text (Windows, Mac, & Linux).

Once you've done that, you should be able to double-click the `.bat` file you made to run your server!

Bam! Bukkit server up and running! It should look something like Figure 10.



```

C:\WINDOWS\system32\cmd.exe
11:02:12 [INFO] Preparing spawn area: 59%
11:02:13 [INFO] Preparing spawn area: 64%
11:02:14 [INFO] Preparing spawn area: 72%
11:02:15 [INFO] Preparing spawn area: 80%
11:02:16 [INFO] Preparing spawn area: 87%
11:02:17 [INFO] Preparing spawn area: 95%
11:02:17 [INFO] Preparing start region for level 1 (Seed: 1250221134302686140)
11:02:19 [INFO] Preparing spawn area: 8%
11:02:20 [INFO] Preparing spawn area: 18%
11:02:21 [INFO] Preparing spawn area: 25%
11:02:22 [INFO] Preparing spawn area: 33%
11:02:23 [INFO] Preparing spawn area: 46%
11:02:24 [INFO] Preparing spawn area: 59%
11:02:25 [INFO] Preparing spawn area: 74%
11:02:27 [INFO] Preparing spawn area: 82%
11:02:28 [INFO] Preparing spawn area: 89%
11:02:29 [INFO] Preparing spawn area: 97%
11:02:29 [INFO] Preparing start region for level 2 (Seed: 1250221134302686140)
11:02:30 [INFO] Preparing spawn area: 30%
11:02:31 [INFO] Preparing spawn area: 57%
11:02:32 [INFO] Preparing spawn area: 91%
11:02:33 [INFO] Done (<30.641s>! For help, type "help" or "?"
11:02:36 [WARNING] Can't keep up! Did the system time change, or is the server overloaded?
>

```

Fig. 10

You might have noticed that the folder you put the `.jar` file in is now populated with a LOT of new folders and files. The important ones are still there; “world”, “server.properties”, etc. But there is one new folder and a new file you will need to consider.

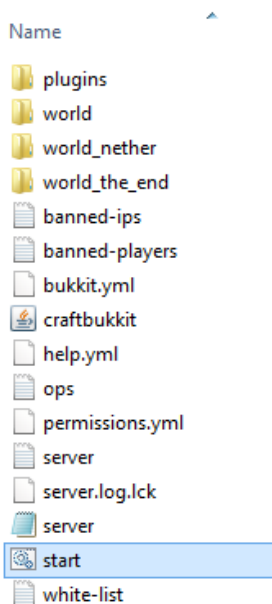


Fig. 11

The **plugins** folder is where you're going to store all of your Bukkit plugins. Pretty self-explanatory, yes? The great thing about Bukkit is how easy it is to add and remove plugins. When you download a plugin (plugins are also `.jar` files) from [plugins.bukkit.org](http://plugins.bukkit.org), just move it into the **plugins** folder and reboot your server!

The **permissions.yml** starts off as an empty file, but this is where you program in which players have permission to use which commands. For the majority of people you will never have to type anything into this file, but it is important to know that it's there and what it does. The server I administrate uses this file to delegate *some* OP commands to hopeful future Admins. We give them some power and see if they can be trusted with it for a while before giving them full-blown access to all of the OP commands.

## Minecraft Version

Here's where things get a bit hairy. Bukkit servers will usually require that any player that joins to be playing a specific version of Minecraft. The Minecraft version you will need is dictated by the version of Bukkit you're running, and your Bukkit's version is dictated by which plugins you are using.

Information about the current release of Bukkit can be found on their main page at [bukkit.org](http://bukkit.org). The news stories on the home page will announce which version is the most recent (see Figure 12), and their version numbers coincide with which version of Minecraft you should use. In Figure 12, you can see that it's CraftBukkit 1.7.9, so we know that this version will work with Minecraft 1.7.9.



Fig. 12

You can find information about which version the plugin supports on the plugin's main page on [dev.bukkit.org](http://dev.bukkit.org) in the right-hand sidebar under the heading "Facts". Shown in Figure 13.

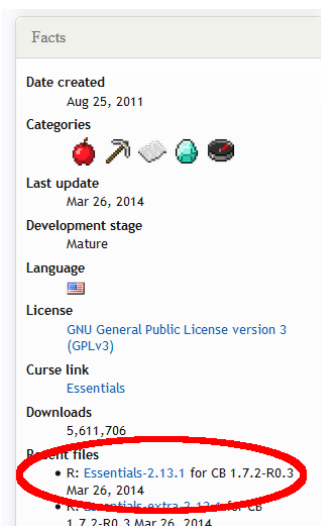


Fig. 13

As an example: If I want to use the Essentials plugin (which is the plugin whose data is shown in Figure 13), I can see that it requires Bukkit (aka CraftBukkit, or CB as it is stated there) version 1.7.2. Since the Bukkit version number coincides with the same Minecraft version number, we know that we'll need to use Minecraft version 1.7.2 in order to play on this server.

Yeah, it's weird, but there you go. That's what you get for being interested in a game with such an expansive modding community. Luckily, we can easily change which version of Minecraft we're using. When you launch Minecraft, there are a few tabs at the top of the launcher. One is titled "Profile Editor". Click on it, and then double-click the line that has your current user on it. Check out Figure 14 if you're confused.

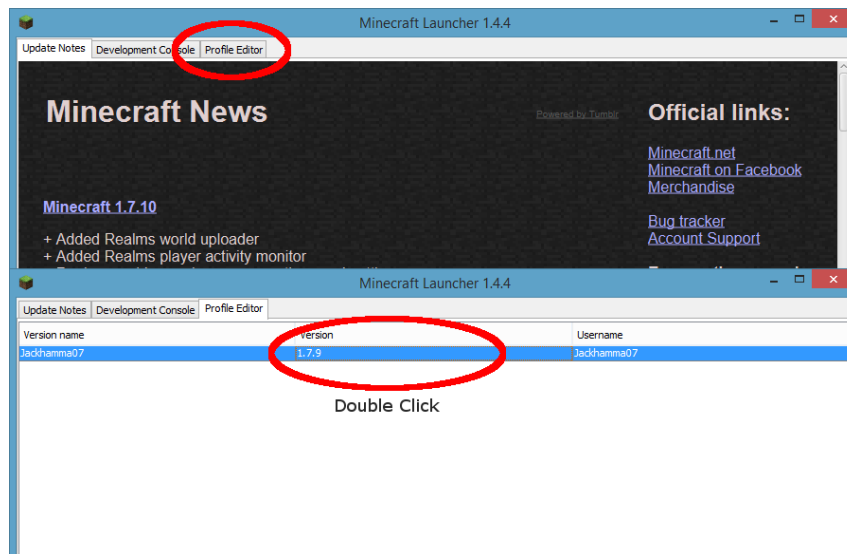


Fig. 14

In the Profile Editor window (towards the bottom) there is a drop-down menu titled “Use version:” Use this menu to choose which Minecraft version you are going to play with.

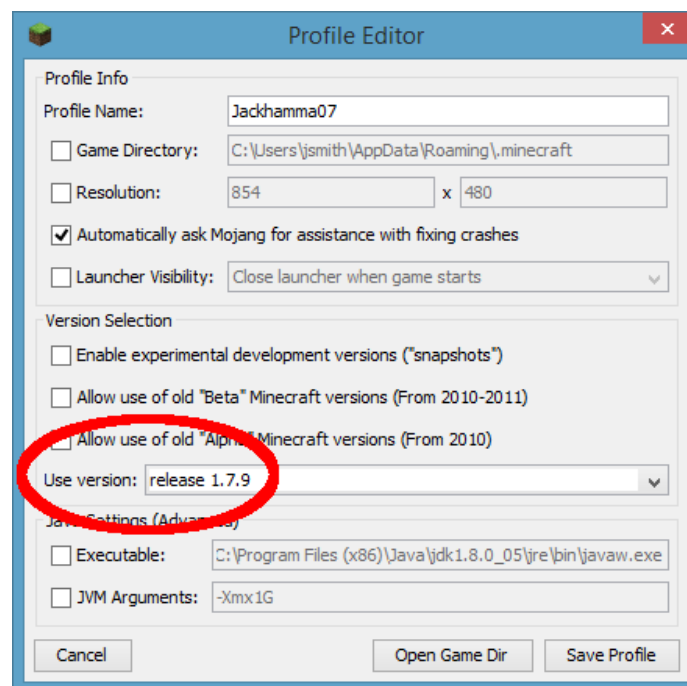


Fig. 15

Once done, click “Save Profile” and then “Play” like you normally would. It might take a bit longer to launch this time since Minecraft will need to download an entire other version for you to play on. This will only have to happen once, though.

Now that that’s out of the way, let’s get back to Bukkit. Here’s how to download, install, and use a particularly useful plugin.

## Plugin Example

There are thousands of plugins you could use for your own Bukkit server. To demonstrate how to install and use a plugin, I'm going to use the Essentials plugin found at [dev.bukkit.org/bukkit-plugins/essentials/](http://dev.bukkit.org/bukkit-plugins/essentials/). As stated before: I **strongly** recommend you do some research on any and all plugins you plan on using on your server. This plugin is no different. Take some time to read through the documentation and understand what it does before you hit the download button shown in Figure 16.

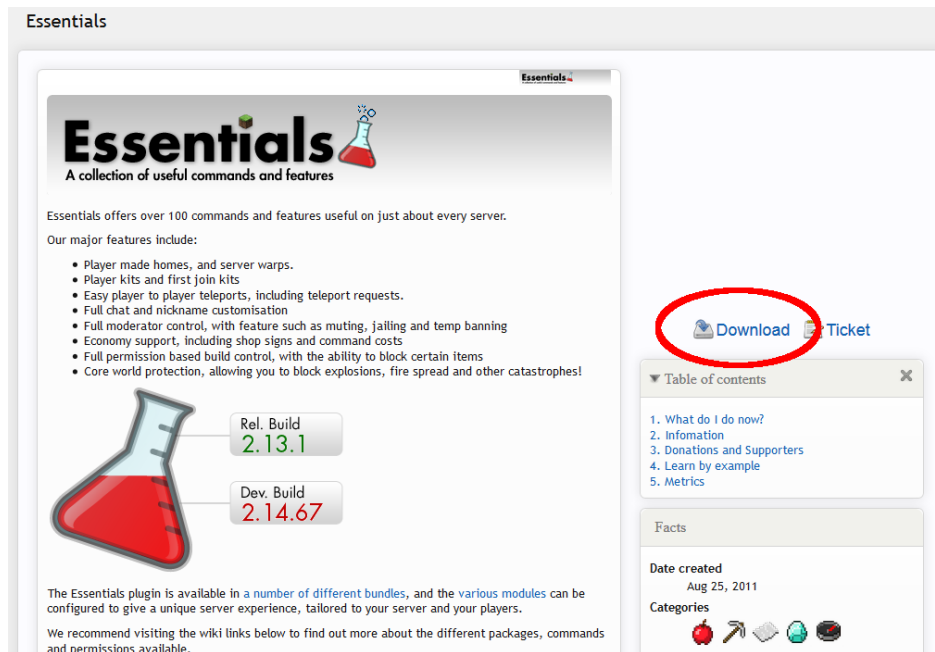


Fig. 16

Figure 13 (a few pages back) shows us the “Facts” sidebar regarding essentials. We can see that the most recent reliable release of Essentials is 2.13.1 (the developer build - dev build - is a less reliable, but more up-to-date version. I don't recommend using the dev builds on your public servers). Version 2.13.1 is compatible with Bukkit 1.7.2, so that is the version we'll have to run on our server as well as the version of Minecraft we'll have to run on our own PCs when we want to play. By clicking the download link circled in Figure 16, we'll download a .jar file containing the plugin. When this finishes, all we need to do is drag and drop (or copy and paste) this .jar file into the “plugins” folder shown in Figure 11.

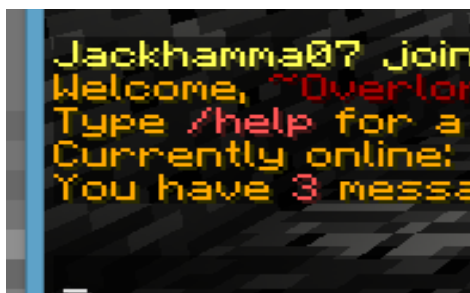


Fig. 17

Ta-da! It's installed! Now all you have to do is reboot your server and log into Minecraft, and we can get into actually using it. Once you log into your server via Minecraft, press the “T” key to bring up the chat box. You should see a little blinking cursor (shown in Figure 17). This is where you put in every command as an OP as well as chat with other players on your server.



Fig. 18

are added with the Essentials plugin is massive and range from important functions like repairing weapons and spawning Kits to adding commands to make a simple Economy on your server.

Most plugins also have in-game help files that will assist you in navigating through all of the possible commands. As shown in Figure 18, I can type “/help” and get a small list of important commands (shown in Figure 19).

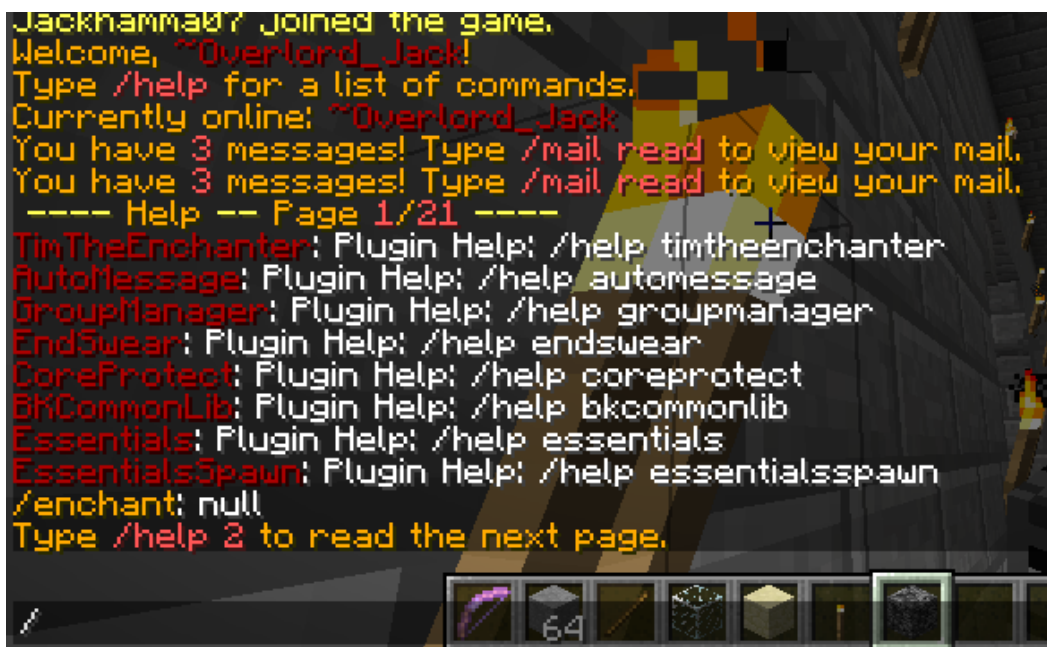


Fig. 19

We can see from Figure 19 that I have a few other plugins installed on this particular server, and that there are 20 more pages of commands that I can look through. On top of that, each of the other plugins (listed in red) have their own set of commands, and you can find them by invoking their own “Help” command (listed next to the red entries).

Every OP command is preceded by a forward slash ( / ). By using this slash, the server understands that you’re trying to perform a certain command. Since you have already perused the Essentials documentation, you already know what will probably happen if you type “/spawn”...

Nothing? Well, that particular command is meant to teleport its invoker to the server’s spawn point. If you haven’t moved from the spawn point, then you might not perceive a change. The list of commands that



## Administrating Your Server

This section is the most philosophical part of the document. Not everyone is going to run their server the same way. Some people will try to rule with an iron fist while others invoke a gentle touch. For me, I try to be as hands-off as possible. I view OPs as server police, and as such aren't supposed to help players cheat the system by spawning in gobs of diamonds or by throwing other players into lava. If something goes wrong, or if a player is being a nuisance, OPs are there to remedy the situation. There are plenty of public Minecraft servers for people to play on, so I try to avoid doing anything that would drive players away. However, not every policy I may put in place will be popular, which is a good reason to take things slowly. Don't regenerate the server's world without proper warning to your regular players, don't randomly clear players' inventories, etc. You're there to facilitate other players' enjoyment while they're playing the game with their friends.

Naturally, you might not agree with me, and that's totally fine. Run the server how you like, and learn by trial and error. I feel there are a few things that you truly should keep in mind, though:

1. Know more about how your plugins work than your players.
2. "If you give a mouse a cookie..."
3. Your OPs should be people you trust and people you can readily contact.
4. Severely limit the number of people who can access your server files.
  - a. Seriously. Like... limit it to one person if possible.
5. Patience. Always try patience, but don't be afraid to ban someone.

I feel these guidelines are foundational and are essential for beginning server admin and OPs. There will always be troublemakers, but then again there will be players that genuinely enjoy the server and appreciate the service you're giving them.