

# THE ULTIMATE GUIDE TO MINECRAFT® SERVER



TIMOTHY L. WARNER

que

UNOFFICIAL GUIDE

FREE SAMPLE CHAPTER

SHARE WITH OTHERS



The Ultimate Guide to

# MINECRAFT® SERVER

Timothy L. Warner

**que®**

800 East 96th Street,  
Indianapolis, Indiana 46240 USA

# The Ultimate Guide to Minecraft Server

## Copyright © 2016 by Que Publishing

All rights reserved. No part of this book shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission from the publisher. No patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained herein.

ISBN-13: 978-0-7897-5457-8

ISBN-10: 0-7897-5457-6

Library of Congress Control Number: 2015939394

Printed in the United States of America

First Printing: July 2015

## Trademarks

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Que Publishing cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

Minecraft is a trademark of Mojang Synergies / Notch Development AB. This book is not affiliated with or sponsored by Mojang Synergies /Notch Development AB.

## Warning and Disclaimer

Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied. The information provided is on an “as is” basis. The author and the publisher shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book

## Special Sales

For information about buying this title in bulk quantities, or for special sales opportunities (which may include electronic versions; custom cover designs; and content particular to your business, training goals, marketing focus, or branding interests), please contact our corporate sales department at [corpsales@pearsoned.com](mailto:corpsales@pearsoned.com) or (800) 382-3419.

For government sales inquiries, please contact [governmentsales@pearsoned.com](mailto:governmentsales@pearsoned.com).

For questions about sales outside the U.S., please contact [international@pearsoned.com](mailto:international@pearsoned.com).

### Editor-in-Chief

Greg Wiegand

### Executive Editor

Rick Kughen

### Development Editor

William Abner

### Managing Editor

Sandra Schroeder

### Project Editor

Seth Kerney

### Copy Editor

Cheri Clark

### Indexer

Erika Millen

### Proofreader

Jess DeGabriele

### Technical Editor

John Baichtal

### Editorial Assistant

Cindy Teeters

### Book Designer

Mark Shirar

### Compositor

Bronkella Publishing

---

## Contents at a Glance

	Introduction	1
<b>CHAPTER 1</b>	Minecraft Multiplayer from the Gamer's Perspective	5
<b>CHAPTER 2</b>	Building a Vanilla Minecraft Server	25
<b>CHAPTER 3</b>	Operating a Vanilla Minecraft Server	45
<b>CHAPTER 4</b>	Understanding Networking as It Relates to Minecraft	71
<b>CHAPTER 5</b>	Installing a Custom Minecraft Server	93
<b>CHAPTER 6</b>	Integrating Mods into Your Minecraft Server	113
<b>CHAPTER 7</b>	Exploring Minecraft Realms	137
<b>CHAPTER 8</b>	Taking Control of Minecraft with Third-Party Hosting	159
<b>CHAPTER 9</b>	Giving Your Players the Best Gaming Experience	185
<b>CHAPTER 10</b>	Monetizing Your Minecraft Server	209
<b>APPENDIX</b>	Tim's "Top 10" Lists	237
	Index	247



# Table of Contents

## Introduction   1

Who Should Read This Book   1

How This Book Is Organized   2

Conventions Used in This Book   3

    Follow Me!   3

    About the Bitly Hyperlinks   3

System Requirements   4

## Chapter 1   **Minecraft Multiplayer from the Gamer's Perspective   5**

Understanding Minecraft Multiplayer   6

    Minecraft Demo Mode Versus Minecraft Premium   7

    Please Avoid Cracked Launchers   8

    The Three Types of Minecraft Multiplayer   10

Finding “Good” Minecraft Servers   10

    Understanding Minecraft Server Lists   11

    A Word on Minecraft Minigames   14

    About Connection Addresses   15

    Public Versus Whitelist Servers   15

Joining an Online Server   16

Minecraft Multiplayer Netiquette   19

    The Importance of Help   19

    Chatting with OPs and Other Players   20

    Setting Chat Options   22

    The Three Rules of Multiplayer Netiquette   23

*Rule #1: Obey the Server Rules—Or Leave*   23

*Rule #2: Act in Accordance with the “Golden Rule”*   23

*Rule #3: Ask Without Fear*   24

The Bottom Line   24

## Chapter 2 Building a Vanilla Minecraft Server 25

### Preparing the Java Environment 25

Verifying Your Java Version (Windows) 26

Verifying Your Java Version (OS X) 27

Downloading and Installing the JRE 27

### Installing the Minecraft Server 28

The Windows Installation Process 29

The OS X Installation Process 31

### An Interlude...Your Lab Environment 34

Buy Another Copy of Minecraft? Really? 34

### First Run and Initial Configuration Steps 34

Understanding the Server Configuration Files 35

The Minecraft Server Console 37

Running More Than One Server Instance 42

### Finding Vanilla Servers on the Internet 43

### The Bottom Line 44

## Chapter 3 Operating a Vanilla Minecraft Server 45

### Understanding Our Test Environment 45

### Configuring the Server's World 47

Introducing Sublime Text 47

Understanding `server.properties` 48

### Getting Your User(s) Connected 50

Scanning, Scanning... 51

### Interacting with Your Users 53

`/say`, `/tell`, and `/me` 55

`/tp` and `/give` 59

### Exerting Discipline on Your Server 59

`/kick` 60

`/ban` and `/ban-ip` 61

`/pardon`, `/pardon-ip`, and `/whitelist` 63

### Adding a Management Layer to Your Server 65

Introducing McMyAdmin 65

Installation Notes 67

Configuration Notes	67
McMyAdmin Editions	68
Opening a Single-Player Game to the LAN	68
The Bottom Line	70

## **Chapter 4   Understanding Networking as It Relates to Minecraft   71**

Revisiting Our Test Network	72
Learning as We Go: Basic Networking Terminology	73
IP Addresses	73
Private and Public Addresses	74
Network Address Translation	75
DHCP	75
Taking the Next Step: Preparing Our Network for Minecraft Server	76
Discovering Our Computer's Networking Configuration	76
Viewing Our Router's Configuration	78
Allowing Minecraft Traffic Through the Firewall	81
Ports	81
Configuring Port Forwarding	83
Testing the Connection	85
Let's Play!	86
Using a Hostname Instead of an IP Address	88
Understanding DNS and DDNS	88
Setting Up No-IP	88
Sweeping Up the Shavings	90
The Bottom Line	91

## **Chapter 5   Installing a Custom Minecraft Server   93**

Beginnings: It's a Wild, Wild West	93
If Not Bukkit, Then What?	94
Installing SpigotMC	95
Starting SpigotMC Server	96
Configuring SpigotMC	97
<code>server.properties</code>	98
<code>spigot.yml</code>	98

---

<b>bukkit.yml</b>	98
Testing the Player Experience	99
<b>Introducing Bukkit Plugins</b>	100
Describing the Most Popular Plugins	100
A Question of Balance	102
<b>Installing, Configuring, and Using Bukkit Plugins</b>	102
Obtaining Essentials: RTFM	102
Configuring Essentials	106
Testing the Gameplay Experience	108
<b>The Bottom Line</b>	112
<b>Chapter 6 Integrating Mods into Your Minecraft Server</b>	<b>113</b>
<b>Introducing Cauldron</b>	113
Minecraft Forge in a Nutshell	114
Installing the Forge Client	115
<b>Getting Cauldron Up and Running</b>	117
<b>First, Our Plugin Proof of Concept</b>	118
PermissionsEx Quick Start	118
Testing the Plugin	120
<b>Now for the Good Stuff—Installing a Mod</b>	122
Installing RailCraft	122
Testing the Railcraft Player Experience	125
<b>Experimenting with Integrated Modpacks and Launchers</b>	129
Feed the Beast (FTB)	130
Installing FTB Server	132
Starting the FTB Launcher	134
ATLauncher and Other Options	134
<b>Quality Client Mods</b>	135
<b>The Bottom Line</b>	136
<b>Chapter 7 Exploring Minecraft Realms</b>	<b>137</b>
<b>Understanding Cloud Services</b>	137
Pros and Cons	139
<b>Introducing Minecraft Realms</b>	139
Let's Sign Up!	140

**Playing in a Minecraft Realms World 142**

Inviting Players 145

Managing Players 148

**A Brief Tutorial on Command Blocks 150**

About Target Selectors 151

Programming Command Blocks 152

**Configuring Minecraft Realms 154**

Backup and Restore 154

Viewing Backups 154

Forcing a Backup 155

Downloading a World to Single-Player 155

Uploading a World to Realms 156

Restoring a World 157

**The Bottom Line 157****Chapter 8 Taking Control of Minecraft with Third-Party Hosting 159****What to Look for in a Third-Party Minecraft Host 160**

Minecraft Hosting Terminology 160

What's Important in a Minecraft Host 161

**Joining MCProHosting 163****Getting to Know the Control Panel 168**

Configuring and Starting Our Minecraft Server 169

Logging In 170

Adding Some Plugins 172

**Making Additional Tweaks to Your Server 175**

Editing Your Config Files 175

Deploying a Resource Pack 177

**Mapping Your Server to a Domain 180**

Domain Name Mapping with GoDaddy 180

**A Brief Roster of Other Well-Regarded Minecraft Hosting Companies 183****The Bottom Line 184**

## **Chapter 9 Giving Your Players the Best Gaming Experience 185**

### **Protecting Your Server from Abuse 186**

Distributed Denial-of-Service Attack 186

What to Do to Prevent DDoS Attacks 187

Griefing 188

Hack Yourself 189

Blocking Grieferers 190

Tips for Addressing Grieferers 194

### **Customizing the World Spawn 194**

Setting the Server Spawn 195

Building a Spawn Lobby 196

Preparing the Land for a Lobby Schematic 197

WorldEdit—A More Flexible Terraforming Solution 199

Using WorldEdit to Drop in a Schematic 200

### **Supporting Multiple Worlds and Teleportation 202**

Using Multiverse 202

Creating and Teleporting Between Worlds 203

Deploying Portals 203

Connecting the Two Worlds 206

### **The Bottom Line 208**

## **Chapter 10 Monetizing Your Minecraft Server 209**

### **Understanding Mojang's Position on Minecraft Server Monetization 210**

The New Minecraft EULA 210

The Backlash Against Mojang 211

### **Monetizing Minecraft While Maintaining EULA Compliance 212**

Donations 212

In-Game Advertising and Sponsorship 216

Cosmetic Upgrade Sales 217

VIP Server Access or Subscriptions 219

### **Setting Up BuyCraft 220**

Creating a Paid Package 223

Giving a Paid Item to All Players 225

Creating a Free Package 226

**Advertising Your Minecraft Server 228**

Where to Advertise 228

The Importance of the Server Status Banner 229

Registering Your Minecraft Server 231

About Voting 232

**Someplace to Compare Yourself 233**

Creating a Custom Server Entry for the Minecraft Client 234

Customizing the MOTD 235

**Final Thoughts 235****Appendix Tim's "Top 10" Lists 237**

Top 10 Minecraft Hosts 237

Top 10 Public Minecraft Servers 239

Top 10 Bukkit Plugins 240

Top 10 Modpacks 240

Top 10 Resource/Texture Packs 242

Top 10 Minecraft YouTubers 243

Top 10 Minecraft Minigames 243

Top 10 Minecraft Custom Maps 244

Top 10 Minecraft Seeds 245

**Index 247**

## About the Author

**Timothy Warner** is an IT professional and technical trainer based in Nashville, Tennessee. Tim became acquainted with information technology in 1982 when his dad bought the family a Timex Sinclair 1000 home computer and he taught himself BASIC programming. Today he works as an author/evangelist for Pluralsight and shares Windows PowerShell knowledge with anyone who'll listen at his Two Minute PowerShell blog: <http://2minutepowershell.com>. You can reach Tim directly via LinkedIn: <http://linkedin.com/in/timothywarner>.



## Dedication

*To my favorite Minecraft buddies: Xander, Xan, Luther, Daisy, and Zoey. Watch out for those Creepers!*

## Acknowledgments

Thanks to Markus Persson and the Mojang team for giving the world Minecraft. You guys must know in your hearts the huge impact the game has had on all of us. Thanks also to the entire third-party Minecraft community—you folks inspire me to be a better gamer, a better sysadmin, and a better programmer.

It may take a village to raise a child, but I know that it takes a large office full of talented professionals to publish a book. To that end, I want to thank my editor Rick Kughen for never doubting my abilities even when I doubted them myself. Thanks to my publisher, Greg Wiegand, for being so willing to embrace my sometimes wacky ideas.

Editorial and production staff rarely receive the credit they deserve. Thanks so much to John Baichtal, my technical editor, and to Cheri Clark, my copy editor, for making my writing as clean and accurate as it can be. I extend my gratitude as well to Seth Kerney, my production editor, and to the ever-helpful Kristen Watterson for streamlining the book publishing process.

Thanks to my family and friends for your love and encouragement. Finally and most importantly, thank you, my reader! I hope that this book takes your Minecraft gaming to the next level, and possibly serves as a springboard for new hobbies and career opportunities.

## We Want to Hear from You!

As the reader of this book, *you* are our most important critic and commentator. We value your opinion and want to know what we're doing right, what we could do better, what areas you'd like to see us publish in, and any other words of wisdom you're willing to pass our way.

We welcome your comments. You can email or write to let us know what you did or didn't like about this book—as well as what we can do to make our books better.

*Please note that we cannot help you with technical problems related to the topic of this book.*

When you write, please be sure to include this book's title and author as well as your name and email address. We will carefully review your comments and share them with the author and editors who worked on the book.

Email: [feedback@quepublishing.com](mailto:feedback@quepublishing.com)

Mail: Que Publishing  
ATTN: Reader Feedback  
800 East 96th Street  
Indianapolis, IN 46240 USA

## Reader Services

Visit our website and register this book at [quepublishing.com/register](http://quepublishing.com/register) for convenient access to any updates, downloads, or errata that might be available for this book.

*This page intentionally left blank*



# Introduction

**“Success is a journey, not a destination. The doing is often more important than the outcome.”**

—Arthur Ashe, legendary professional tennis player

Hello, and welcome to Minecraft server training! I invested dozens of hours in the Minecraft single-player game before I found the joy of multiplayer. Because I’ve been a fan of first-person shooter (FPS) games since *Wolfenstein 3D* (remember that one?), I enjoy player vs. player (PvP) matches in shared Minecraft worlds.

I’ve also been a bit of a nerd regarding role-playing games (RPGs); again, I can have this in Minecraft multiplayer by exploring and building with friends from all over the world.

This book, however, isn’t so much about the multiplayer player’s experience (although I devote a chapter to that subject). Instead, I’m going to teach you everything you need to know to build and maintain your very own Minecraft server, from soup to nuts. Are you excited? You should be!

## Who Should Read This Book

As an author, I always write with my three primary audiences in mind:

- **Potential Minecraft Server Admins:** Perhaps you’ve experimented with hosting a Minecraft server and gotten bogged down in all the details. *Hint:* Most YouTube “instructors” don’t know how to instruct at all. You’re reading this book so that you can get clear, easy-to-follow directions to set up an awesome Minecraft server.
- **Geeky Types:** You’re the kind of person who digested my *Hacking Raspberry Pi* book (<http://www.amazon.com/Hacking-Raspberry-Pi-Timothy-Warner/dp/0789751569>) in one sitting. You intend to leverage your new Minecraft server skills to become familiar with computer networking and network security for other purposes (maybe building a new career!).

- **Teachers and Students:** I'm both a teacher and a student, so I know more than most about how useful Minecraft is as an educational tool. You can learn to do computer programming by building Minecraft mods and Bukkit plug-ins. You can learn how to set up and manage computer networks by hosting your own Minecraft server.

If you find that you don't belong in any of the previous three classifications, don't worry about it. Set your sights on learning as much as you can and, above all else, having fun, and you'll be fine!

## How This Book Is Organized

I always write books in such a way that you'll derive maximum benefit by reading the chapters in order. With no further buildup, allow me to present the chapter-by-chapter details on how I structured the content:

- Chapter 1, "Minecraft Multiplayer from the Gamer's Perspective," brings new Minecraft multiplayer gamers up to speed with how the game works and how to navigate within its worlds.
- In Chapter 2, "Building a Vanilla Minecraft Server," you learn how to use Mojang's own authorized Minecraft server. I give you instructions for both Windows and OS X systems.
- In Chapter 3, "Operating a Vanilla Minecraft Server," you take your understanding of the official Minecraft server to the next level. Although the platform is limited, take heart because you'll apply all your newfound skills to custom servers later in the book.
- In Chapter 4, "Understanding Networking as It Relates to Minecraft," you master the fundamentals of data networking. This is a huge subject, to be sure. However, I give you what you need to host your own Minecraft server with confidence.
- In Chapter 5, "Installing a Custom Minecraft Server," you move off the Mojang reservation and embrace Minecraft server third-party development. You'll get up to speed with what's what in the often-confusing and ever-volatile world of Bukkit, Spigot, and other server platforms.
- In Chapter 6, "Integrating Mods into Your Minecraft Server," you begin to understand how we can broaden and deepen our Minecraft server's feature set by deploying mods. Again, this can be a confusing subject to tackle on your own, but you're in good hands with me!
- In Chapter 7, "Exploring Minecraft Realms," you learn how to use Mojang's "other" officially licensed server. Realms is a cloud-based service, which means that you're saved from most of the back-end configuration that you're responsible for when you host your server inside your home.



- In Chapter 8, “Taking Control of Minecraft with Third-Party Hosting,” you learn how you can combine the flexibility of third-party unofficial Minecraft servers with the stability, security, and reliability of a cloud server such as Realms.
- In Chapter 9, “Giving Your Players the Best Gaming Experience,” you pick up best-practice tips for staying focused on your player base, avoiding server damage through griefing, and ensuring that players want to stay on your server for a long time to come.
- In Chapter 10, “Monetizing Your Minecraft Server,” you learn what options you have for recouping server costs while staying within the bounds of the Mojang end user license agreement (EULA).

## Conventions Used in This Book

In my experience as an author and a teacher, I’ve found that many readers and students skip over this part of the book. Congratulations for reading it! Doing so will pay off in big dividends because you’ll understand how and why we formatted this book the way that we did.

### Follow Me!

Throughout the book, you’ll find “Follow Me!” exercises, which are opportunities for you to apply what you’re learning right then and there in the book. I do believe in knowledge stacking, so you can expect that later “Follow Me!” exercises assume that you know how to do stuff that you did in previous exercises.

Therefore, your best bet is to read each chapter in sequence and work through every “Follow me!” exercise.

### About the Bitly Hyperlinks

Whenever I want to point you to an Internet resource to broaden and deepen the content you’re learning, I provide a uniform resource locator (URL, also called an Internet address) in the following form:

`http://bit.ly/uaKpYD`

You might wonder what the heck this is. The way I look at it, if I were reading this title as a print book and needed to type out a URL given to me by the author, I would rather type in a “shortie” URL than some long, crazy URL with all sorts of special characters. *The most important thing I have to tell you concerning the bit.ly short URLs is that the ending part is case sensitive.* Therefore, typing the previous URL as, say, `http://bit.ly/UaKpyD` isn’t going to get you to the same page as what I intended.

## System Requirements

You don't need a heck of a lot, computer-wise, to work through all the procedures I provide in this book. Let me give you the run-down:

- **A Windows or Mac computer:** As I'm sure you know, Minecraft runs on just about every hardware platform known to humankind. However, I've found the multiplayer experience in the "classic" Windows and OS X versions to be the most flexible and full-featured. As I said earlier, I give you all instructions for both Macs and Windows computers.
- **An Internet connection:** This requirement should be a no-brainer. After all, how can you host an Internet-accessible Minecraft multiplayer server unless you have a live connection to the Internet!
- **Basic computer navigational skills:** If you're worried about having to know Java programming to learn Minecraft server, don't be. We aren't developing Minecraft mods here, after all. That said, you'll need to know how to move around your computer, create folders, copy files, and create, edit, and save text files.

Okay—that's enough of the preliminaries. It's time to learn how to do Minecraft multiplayer!



**“Fools ignore complexity. Pragmatists suffer it. Some can avoid it. Geniuses remove it.”**

**—Alan Perlis, American computer scientist and first recipient of the Turing Award**

## **Understanding Networking as It Relates to Minecraft**

What You’ll Learn in This Chapter:

- How TCP/IP networking relates to Minecraft multiplayer
- How to view your computer and router IP settings
- How to put your home-based Minecraft server on the Internet
- How to use a friendly hostname for your server instead of an unfriendly IP address

I’m not sure if you’re aware of it, but Minecraft is being used all over the world, every day, as an educational tool. Gaming as a teaching and learning method isn’t exactly new; however, Minecraft is such a rich and varied game that you can approach not only gameplay, but the game itself in many different directions.

One of my goals in writing this book is to generate interest in my readers, especially my younger readers, in systems administration. The IT field has been spectacular to me since I entered it in 1997, and there’s room for you too!

If you’re interested in discovering the magic of computer programming, developing Minecraft mods might be just the ticket to get you started. In this chapter we cover computer networking (a huge topic) in the space of only a couple dozen pages.

By the end of the chapter, you not only will understand what you need to do to put your Minecraft LAN server on the public Internet to host external players, but also will know the “hows” and “whys” behind your configuration. And who knows—you might discover that you have enough passion and aptitude for server administration that you might want to make a career of it!

Let’s begin!

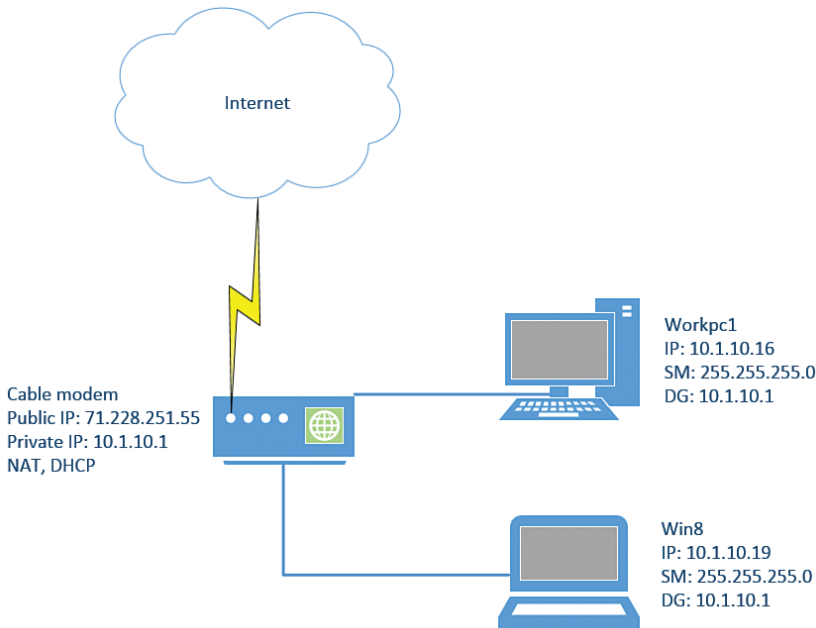


## Revisiting Our Test Network

I want to share another network diagram with you that shows the topology we'll be working with. "Topology" is a networking term that simply refers to the physical arrangement of computers and associated devices on a single network.

You can see the topology in Figure 4.1. Let me describe the major working parts:

- Workpc1 is a Windows 8.1 desktop computer that hosts a vanilla Minecraft server. The computer receives its IP address from the cable modem.
- Win8 is a Windows 8.1 laptop computer that, like Workpc1, obtains its IP address from the cable modem.
- The cable modem is directly connected to the Internet. The device has an Internet-connected IP address and an internal, private interface that connects to Workpc1 and Win8. The router (because that's actually what the cable "modem" is) performs Network Address Translation (NAT) and Dynamic Host Configuration Protocol (DHCP) services on behalf of Workpc1 and Win8.



**FIGURE 4.1** The home-based network that we'll use as a case study in this chapter.

# Learning as We Go: Basic Networking Terminology

Rather than give you dry, boring explanations of the basic networking vocabulary terms, we'll simply use the terms in context as we work with our network devices.

You'll notice that I use the term *network devices* (or hosts) instead of *computers*. I do this because on a TCP/IP network, any electronic device that has a network interface card (NIC) installed is considered a node, or device, on that network. By this definition, any of the following pieces of hardware can be considered to be network hosts:

- Laptop computer
- Smartphone
- Tablet (iPad, Microsoft Surface, and so on)
- Wi-Fi router
- Smart TV
- Raspberry Pi
- Smart watch

TCP/IP stands for Transmission Control Protocol/Internet Protocol, and it describes a whole bunch of networking protocols, of which TCP and IP are only two. TCP/IP has been the standard networking protocol in the world for the past 20 years or so.

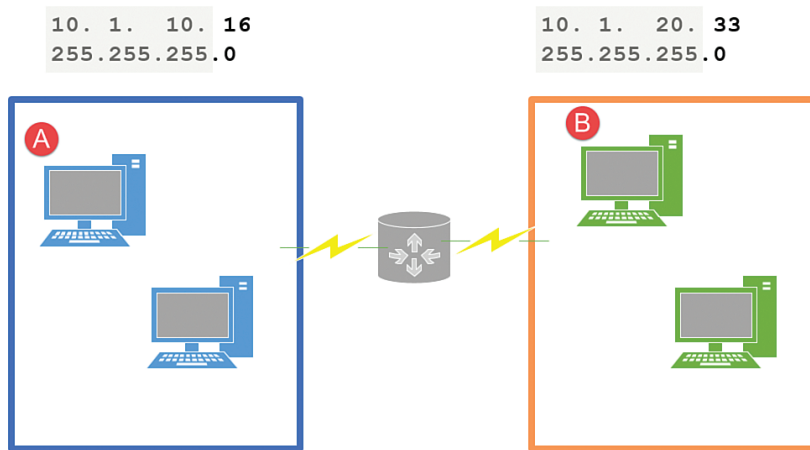
## IP Addresses

For our purposes as Minecraft server administrators, the Internet Protocol (IP) address is the most important concept to master. A networking protocol, such as IP, is simply a set of rules that a TCP/IP host uses to communicate with another host. The sending and receiving of data, in other words.

Specifically, a host's IP address serves to uniquely identify that device on its network. In Figure 4.1, for instance, the cable modem has the IP address 10.1.10.1, Workpc1 has 10.1.10.16, and Win8 has 10.1.10.19.

How do we know that these IP addresses exist on the same network? Besides the fact that the devices are all physically (or wirelessly) connected to the cable modem router, we can take a look at their subnet mask.

The subnet mask is a combination of decimal 255s and 0s that separates the network portion of the address from the host (device) portion. Look at Figure 4.2 to illustrate.



**FIGURE 4.2** TCP/IP hosts use their subnet mask to differentiate their network address as well as host ID.

In Figure 4.2, we have computer A, with IP address 10.1.10.16, attempting to exchange data with computer B, with IP address 10.1.20.33. Computer A “examines” its subnet mask, which “covers” up three of the four parts of an IP address, and determines that 10.1.10 is its own network. Because the target IP address is on network 10.1.20, computer A “decides” it needs to send its traffic to its default gateway (router). The purpose of the router is to serve as a “traffic director” between different IP networks.

To be frank, we don’t need to worry about subnet masks too much for what we’re doing; I just want to make sure that you have a well-rounded understanding.

## Private and Public Addresses

Let’s return to Figure 4.1. If we install Minecraft server on Workpc1, we should have no trouble connecting to the game from Workpc1 because, as we already discussed, the two computers have the same IP network address.

I’ll go so far as to say that if our cable modem had another interface connected to a network address of, say, 10.1.20, then those computers still could connect to our game.

However, if our friend from across town (or on the other side of the world) tries to connect to your Minecraft server with a 10. IP address, then he or she will never connect. Not ever. What’s going on?

It’s a long story, but suffice it to say that the world has run short of public IP addresses that are visible across the Internet. Because of this shortage, routers/cable modems/wireless access points use private IP addresses and two more protocols to bridge the gap.

First, about private IP addresses: You might have noticed that whenever you connect to somebody's Wi-Fi network you receive an IP address from one of the following ranges:

- 10.0.0.0–10.255.255.255
- 172.16.0.0–172.32.255.255
- 192.168.0.0–192.168.255.255

Those IP addresses are called private network addresses because the world has agreed never to route them. You simply cannot send or receive data across the Internet by using a private IP address. Remember what I said about the shortage of public IP addresses; private addresses were developed as a workaround.

## Network Address Translation

You're probably wondering, "If my computer can't communicate on the Internet with a private IP address, then how the heck am I able to surf the Web in the first place?!" The answer is one of those "two additional protocols" I mentioned that your cable modem has: Network Address Translation, or NAT (pronounced like the tiny flying insect).

Your cable modem (or whatever device you use to access the public Internet) has only one public IP address. That's all you get from your Internet service provider (ISP). When I started in IT back in 1997, all computers received public IP addresses from their ISP; those days are long gone now.

With NAT, a router can "share" a single public IP address with more than one device on the internal network. Do you see the beauty here? We can have as many devices inside our network as we want because we're using private IP addressing. All devices connect to the router, and the router serves as a proxy, or go-between, for the Internet and your internal devices.

Of course, some of you are now likely wondering, "How do my internal devices get their private IP addresses? Do they come from the router, my ISP, both, or neither?" That's actually a great question.

## DHCP

Dynamic Host Configuration Protocol (DHCP) is a service for dispensing and managing a range of IP addresses to one or more client devices. It's standard nowadays for your router to serve as a DHCP server. In fact, most routers enable you to edit how DHCP behaves by allowing you to log in to the router directly.

Let's now apply our newfound networking know-how to what's really important: making our Minecraft server available to players from all over the world!

## Taking the Next Step: Preparing Our Network for Minecraft Server

I need to issue a word of warning, especially if you're following along with these procedures in your home network environment. Advertising a vanilla Minecraft server to the Internet from home isn't something you want to do long-term. The reason for this centers on privacy—you likely have more than just Minecraft data on that box. Do you really want to allow anonymous connections from all over the world into your home network and home computer? I didn't think so.

Sure, for testing and learning the technology it's fine, but if you want to do anything more than host a few close friends for some Minecraft fun, then I suggest that you consider not only using an honest-to-goodness Minecraft host, but also using a third-party server instead of the vanilla one.

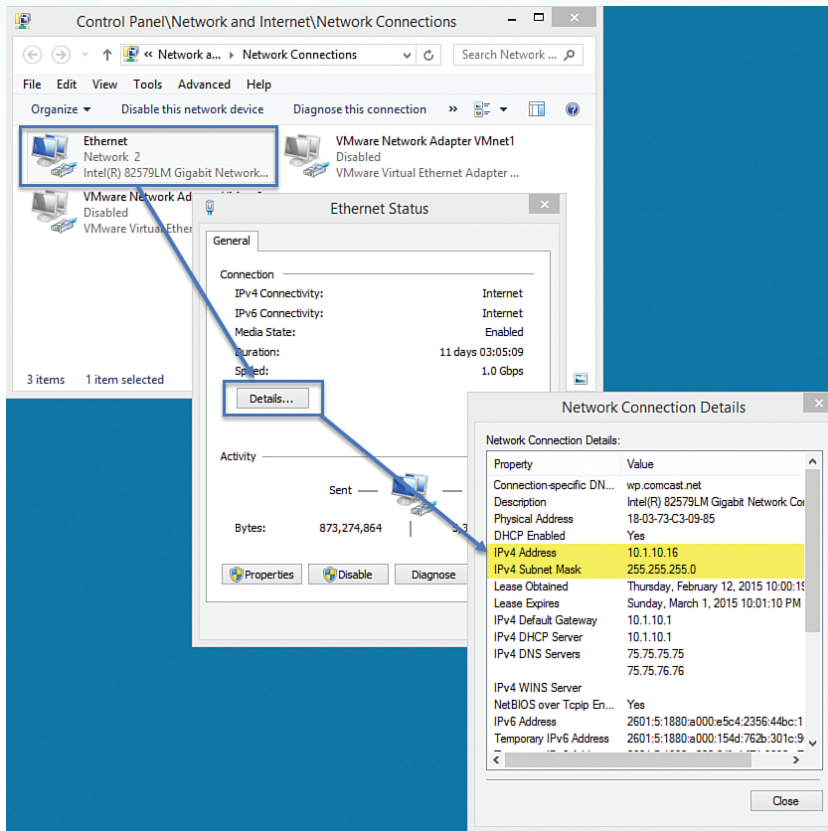
I already mentioned the security and privacy issue as a “deal breaker” for most home-based Minecraft installations. As you work through the rest of this book I'll make an excellent case for using an online hosting service—be patient!

## Discovering Our Computer's Networking Configuration

You can quickly and easily find out what your computer's TCP/IP configuration is either by using graphical tools built directly into the operating system, or by using command-line tools. I'll focus on GUI tools so that we can spend more time analyzing the data and less horsing around with command-line arguments.

On Windows computers, follow this procedure:

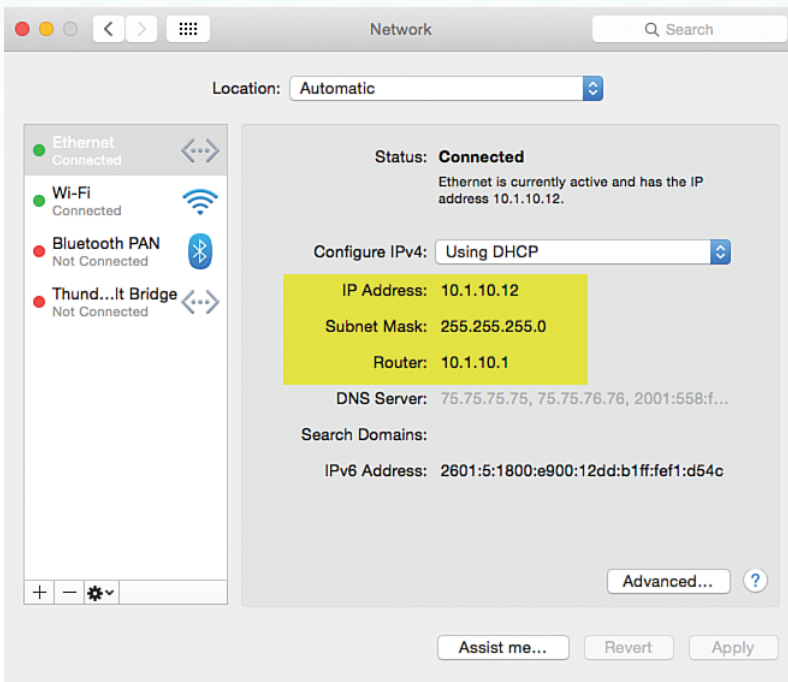
- 1 Press Windows+R to bring up the Run dialog box.
- 2 Type `ncpa.cpl` and press Enter. This command opens your network connections folder and in my experience is much faster than monkeying around with Control Panel.
- 3 In the `Network Connections` folder, double-click the appropriate network interface.
- 4 In the Ethernet Status window, click Details.
- 5 In the Network Connection Details window, shown in Figure 4.3, scan the output, paying attention to the following fields:
  - IPv4 Address
  - IPv4 Subnet Mask
  - IPv4 Default Gateway (this is the router's internal IP address)
  - IPv4 DHCP Server (this should also point to your router's internal IP address)



**FIGURE 4.3** The Windows Control Panel gives us access to all computer TCP/IP configuration settings.

On OS X computers, try the following:

- 1 Open the Apple menu and click System Preferences.
- 2 In the System Preferences pane, click Network.
- 3 In the Network window, shown in Figure 4.4, select your Internet-connected network and scan the results. Pay attention to the following fields:  
 IP Address  
 Subnet Mask  
 Router (this is called “default gateway” in Windows, but the two terms are synonymous)



**FIGURE 4.4** Viewing our network connection details in Apple OS X Mavericks.

## NOTE

### IPv4 Versus IPv6

Do you remember when I mentioned the scarcity of IP addresses? Specifically, I'm referring to IP version 4 (IPv4) addresses, which consist of four decimals between 0 and 255. IP version 6 (IPv6) is the next generation of the IP protocol, and has almost infinitely more public addresses available—the addresses themselves are 128 bits long as compared to 32 bits in IPv4, and the numbers are hexadecimal in IPv6 as opposed to decimal in IPv4.

We don't need to be concerned with IPv6 in this chapter, but since I'm sure you saw plenty of references to IPv6, I thought the subject was worth mentioning.

## Viewing Our Router's Configuration

Now this is the tricky part because there exists a huge variety of router hardware. Are you a cable Internet subscriber, or do you use DSL? Or satellite? Does your Internet router have a

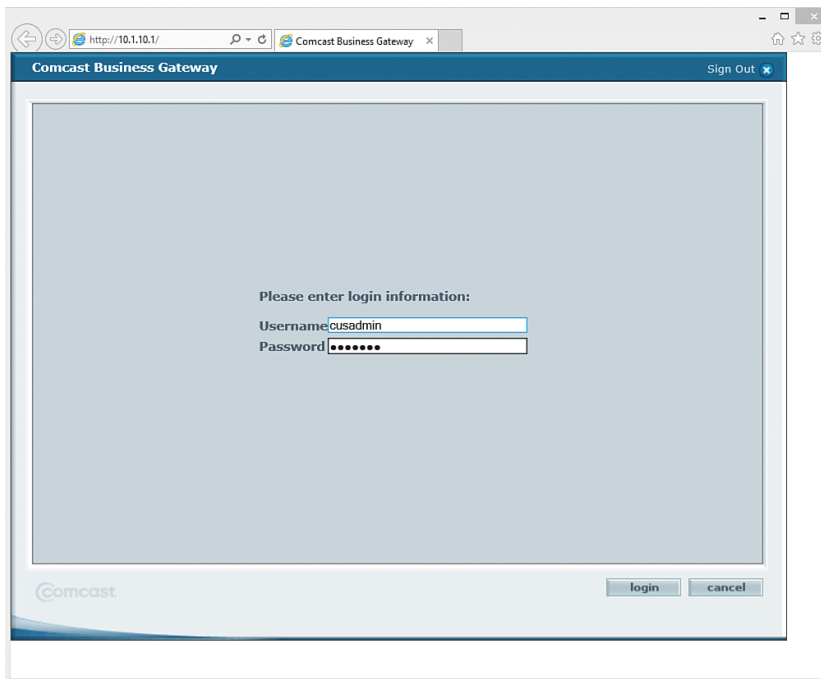
built-in Wi-Fi access point, or have you “daisy-chained” a Wi-Fi router behind your Internet router? Sheesh—so potentially confusing.

Let’s assume that our internal network devices connect to a single Comcast cable modem router. With very few exceptions, you’re allowed to log in to your router directly to make configuration changes. You already know the router’s private IP address; this is your computer’s default gateway address.

Therefore, open a web browser and navigate to that address by using the web standard Hypertext Transfer Protocol (HTTP). For instance, here’s what I need to type in my browser’s address bar to reach my cable modem router:

```
http://10.1.10.1
```

My router’s web-based console and logon screen is shown in Figure 4.5.



**FIGURE 4.5** Most residential routers allow you to log in directly to make configuration changes.



## TIP

If you neither know nor remember your router's login details, one thing you can do before filing a support ticket with your ISP is to check the Web. Sites such as RouterPasswords.com enable you to search on your router's manufacturer and discover the device's default administrative username and password. It's for this reason that you should change your router's password sooner rather than later! Any settings left at their default values are easy targets for malicious users.

After you're in the router's web interface, you can browse around to check settings and make changes. Here are some of the high points:

- Summary of the router's public and private IPv4 addresses
- Details concerning its DHCP setup (shown in Figure 4.6)
- Ability to forward specific traffic to the private internal network from the public Internet

The screenshot shows the Comcast Business Gateway web interface. On the left is a navigation menu with options: MAIN, FEATURE SETTINGS (selected), Administration, LAN, Firewall, Gateway Summary, and HELP. The main content area displays a table of WAN settings and a section for Local IPv4 Settings.

Gateway Status	Network	Cable Modem
WAN Internet IP Address	71.199.206.86	
DNS (primary)	75.75.75.75	
DNS (secondary)	75.75.76.76	
DHCP Time Remaining	62h:36m:47s	
Date	Feb-23-2015	
Static IP Block	20.20.20.1/24	

Local IPv4 Settings	
Gateway IP Address	10.1.10.1
Subnet Mask	255.255.255.0
DHCP Server	Enabled
IP Range (start)	10.1.10.10
IP Range (end)	10.1.10.199

**FIGURE 4.6** Your router's administrative interface tells you exactly how the device operates. Here we see the router's public and private IPv4 addresses.

## Allowing Minecraft Traffic Through the Firewall

Your Internet-facing router acts as a firewall to protect your private internal network. A physical firewall is a divider that separates the passenger compartment from the engine compartment in an automobile; here the word “firewall” takes on a literal meaning.

In a TCP/IP computing context, a firewall is hardware and/or software that by default blocks all incoming Internet traffic to internal devices. The firewall is a good thing, believe me. The problem, if you want to call it that, is that your router’s firewall will block any Internet-based player from accessing your internal Minecraft server. That is, unless you forward the appropriate port(s).

### Ports

TCP/IP ports are cool because they allow your computer, which typically has a single IP address, to participate in all sorts of communication and not get “confused.” This is even more important for your router, which is handling different traffic for different internal devices.

Think of it: You’re browsing [minecraftforum.net](http://minecraftforum.net) from your laptop, while your mom streams YouTube clips to her iPad, while your sister uploads files to her WordPress blog. That’s a lot of network traffic, and it’s all different.

Here’s the deal: Different network services use different default port numbers. Here’s a run-down of some of the most popular, well-known port numbers:

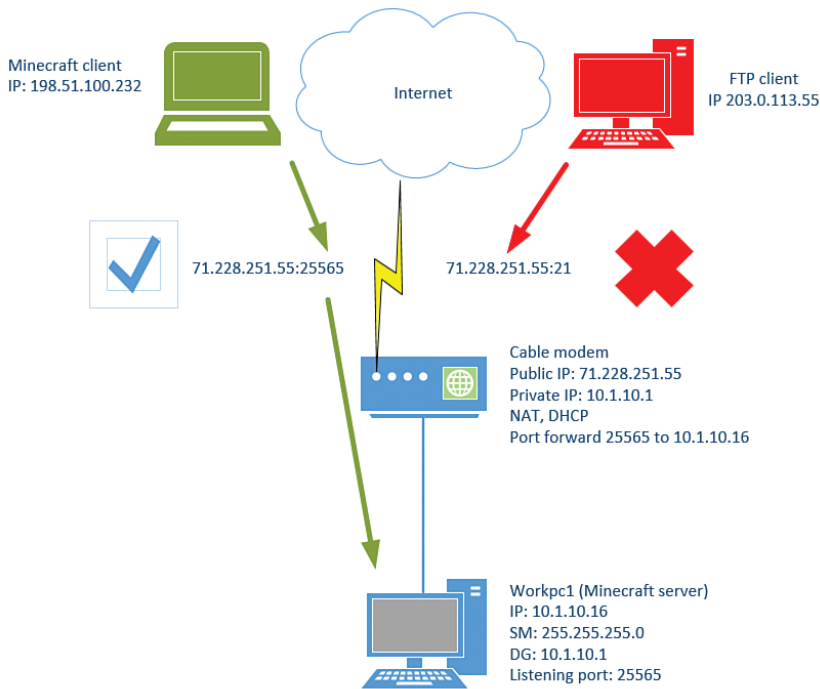
- HTTP (web browsing): 80
- HTTPS (secure HTTP): 443
- FTP (File Transfer Protocol): 21
- SMTP (e-mail): 25
- Minecraft (yeah, buddy!): 25565

Of the previously given ports, the only one I want you to memorize is 25565, because that is Minecraft’s default port number.

What we want to do is forward inbound traffic on port 25565 to our Minecraft server that’s located on the private internal network. Figure 4.7 summarizes port forwarding.

You might be wondering, “How can an Internet player get Minecraft traffic to me when my server has a private IP address? The router has NAT and a public IP address, but the router isn’t the Minecraft server!”

We will give our Internet-based friends our router’s public IP address, and they’ll use that to connect to our Minecraft server. The router’s NAT capability takes care of routing the port 25565 traffic to our server.



**FIGURE 4.7** Schematic diagram showing how port forwarding works.

Let me walk you through Figure 4.7, because I packed a lot of information into that picture:

- Notice that the router has a port-forwarding rule defined such that it allows traffic on port 25565 and forwards it to IP 10.1.10.16, which is our Minecraft server.
- The Internet-based Minecraft client adds a server with the connection address 71.228.251.55:25565 (adding the colon and the port number shouldn't be necessary, but it's wise to use just to be safe) and is successful in connecting to the Workpc1 server.
- The FTP client, also Internet based, is unsuccessful in his or her attempt to establish a File Transfer Protocol (FTP) connection to an internal network resource because (a) the router blocks the traffic because it has no firewall exceptions defined; and (b) we don't have any FTP servers listening for connections in the first place.

I mentioned this in passing, but it bears repeating: You don't have to include the port number to the IP address if the service uses the default port. This is why you can type

`http://yahoo.com`

to reach yahoo.com, instead of this:

```
http://yahoo.com:80
```

Likewise, if your Minecraft server listens on its default port (whose value, you'll recall, is stored as the server-port property in your `server.properties` configuration file), then you shouldn't have to include the port. However, those who want to host more than one Minecraft server on the same box will indeed need to concern themselves with port numbers.

For instance, if I configured port forwarding for a second Minecraft server instance and I used port 25566, then I'd give this address to my Internet-based friends:

```
71.228.251.55:25566
```

## Configuring Port Forwarding

Now let's configure our router for port forwarding!

### TIP

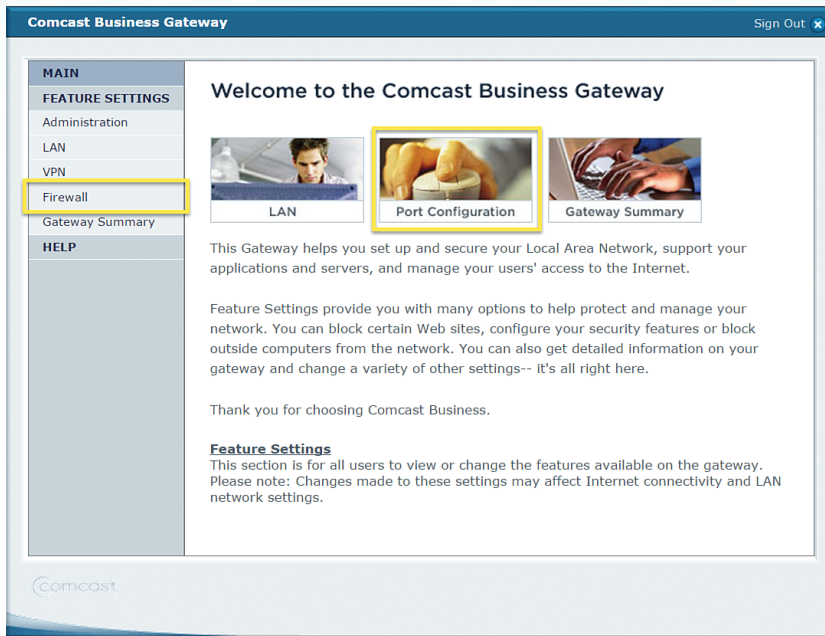
Although the concept of port forwarding is identical no matter what router brand you have, each router manufacturer has a unique web-based management graphical user interface (GUI). Thus, I suggest that you hit up [portforward.com](http://portforward.com) to search for your router brand and receive detailed, step-by-step port-forwarding instructions. If you have a Linksys router, you can visit [ui.linksys.com](http://ui.linksys.com) to play with fully functional router management console simulators.

## FOLLOW ME!

### Configuring Port Forwarding for Our Internal Minecraft Server

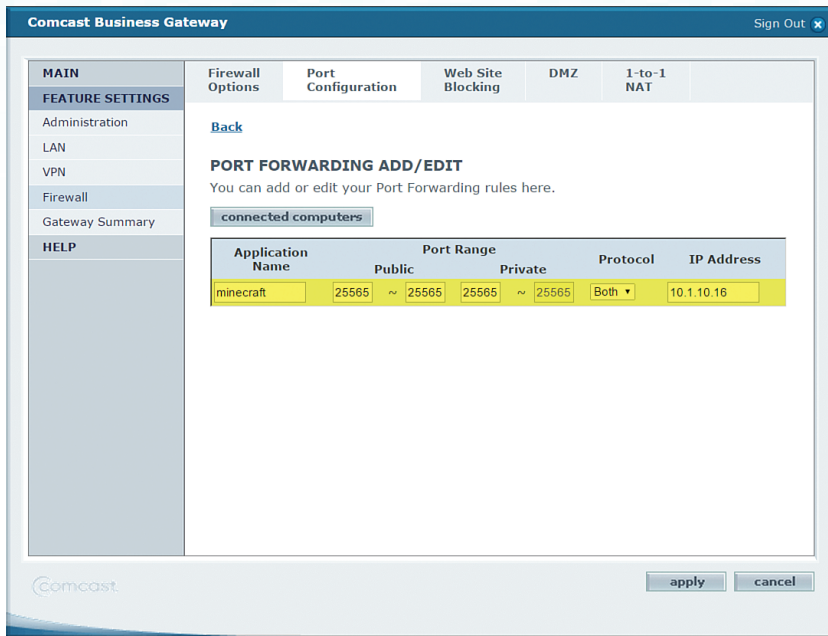
In this exercise we'll configure my Comcast cable modem to forward Minecraft traffic to my Workpc1 computer at IP address 10.1.10.16. You'll need to modify these steps slightly to conform to your particular router and IP addresses.

- 1 Log in to your router as an administrator and navigate to the page on which you can do port forwarding. In Figure 4.8, you can see the home page for my Comcast cable modem. I can click the handy-dandy Port Configuration button to jump directly to the port-forwarding page, or I can click Firewall and then Port Configuration to arrive at the same web page.



**FIGURE 4.8** Router management interfaces sometimes place shortcuts to popular items such as port forwarding right on the home page.

- 2 On my router's Port Forwarding page, we click Add New to define a new port-forwarding rule.
- 3 On the Port Forwarding Add/Edit page, shown in Figure 4.9, I define my rule with the following properties:
  - Name: *Minecraft* (the name doesn't matter; this is just for your own reference).
  - Public: 25565-25565 (the range is to allow you to include more than one contiguous port number in one port-forwarding rule).
  - Private: 25565-25565.
  - Protocol: *Both* (the two choices here are Transmission Control Protocol (TCP) and User Datagram Protocol (UDP); Mojang recommends that you forward both protocols).
  - IP Address: 10.1.10.16 (you'll substitute your Minecraft server's private IP address).
- 4 Click Apply to make the change go into effect.
- 5 On my router, you also have to check the Enable option to activate the port-forwarding rule.



**FIGURE 4.9** It's not as difficult as most Minecraft fans think to forward ports through your router or firewall.

## Testing the Connection

Before we hand out our router's public IP address to our Minecraft buddies, we should check to verify that our router is actually forwarding port 25565 as expected.

### CAUTION

Make sure that you start the Minecraft server on your internal network, or the port-forwarding test will fail. You actually need to have a service online and listening for connections for traffic to make it through your router/firewall.

### TIP

The best way to test port forwarding is to use a computer from outside your home network. Many websites exist that can test port forwarding, but I recommend PortCheckTool.com, shown in Figure 4.10, because it both gives us our public IP address and tests port forwarding.

## PortCheckTool.com - Port Check and IP detection Tool

### Welcome to the Port Check Tool.

Let this tool help you check your ports! Want to know if your server is running? Now you can! Simply enter what port you want to verify into the empty box and click, "Check Your Port". A message will appear, notifying you if your port is blocked by a firewall or ISP.

**Your Current Public IP Address is: 71.228.251.55**

**Success!** I can see your service on 71.228.251.55 on port (25565)  
Your ISP/Router/Firewall is **not** blocking port 25565.

Your IP? 71.228.251.55

What Port? 25565

Check  
Your  
Port

### Common Ports

FTP	21
SSH	22
Telnet	23
SMTP	25
Web	80

**FIGURE 4.10** PortCheckTool.com serves a double duty of (a) giving you your router's public IP address; and (b) verifying that you configured port forwarding correctly.

To use PortCheckTool, simply plug in your router's public IP address in the Your IP field, enter 25565 in the What Port field, and click Check Your Port. As you can see in Figure 4.10, you'll get a "Success!" message if the website can get port 25565 traffic through your router.

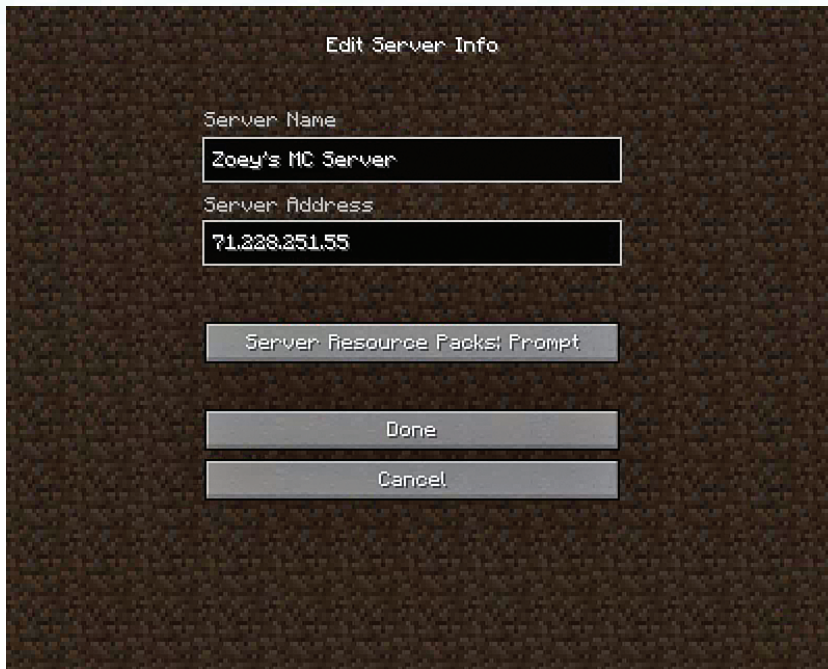
If you get a failure, (a) make sure your Minecraft server is indeed running on your internal network; and (b) log in to your router and verify that the port-forwarding rule is present and enabled.

## Let's Play!

Now run over to one of your Minecraft friends' homes, fire up the Minecraft client, and add your newly published Minecraft server as shown in Figure 4.11.

If all goes well, the newly added server will appear in the server list (see Figure 4.12), and you can join the server's world as usual. Awesomesauce!





**FIGURE 4.11** You can try leaving the :25565 off the server address if you know that the target Minecraft server listens on the default port address.



**FIGURE 4.12** It feels pretty good to see your own Minecraft server in the Minecraft client's server list!



## Using a Hostname Instead of an IP Address

Let's face it—dealing in “raw” IP addresses is tedious. Under the hood, your home router is itself a DHCP client of one of your ISP's DHCP servers. This means that your router's public IP address is likely to change at any time, which will immediately break your home-based Minecraft server.

I would much rather share the Minecraft server connection name

```
timwarner.ddns.net
```

than share this:

```
71.228.251.55:25566
```

Believe it or not, it is both easy and free to make this happen for your home-based Minecraft server. To do this, we need to create an account with a Dynamic Domain Name System (DDNS) service. These are the two companies I recommend:

- No-IP (<http://noip.com>)
- Dyn (<http://dyn.com>)

## Understanding DNS and DDNS

DNS is a TCP/IP network service that translates user-friendly hostnames into IP addresses. For example, if you type

```
http://servers.minecraftforum.net
```

into your browser's address bar, the DNS server with which your computer is associated attempts to resolve the fully qualified domain name (FQDN), `servers.minecraftforum.net`, into the IP address of that particular web server.

The hostname part of the previous FQDN is “servers”; the domain name is “minecraftforum.net.” The specifics of DNS name resolution are far outside the scope of this book; for our purposes, all you need to know is that companies like No-IP and Dyn will map an FQDN to your router's public IP address, and even adjust the mapping on the fly when your router's public IP changes.

## Setting Up No-IP

We'll use No-IP in this example, but all the services work about the same.

---

## FOLLOW ME!

### Associate Your Minecraft Server By Hostname Instead of IP Address

In this exercise, we'll create a free No-IP account and then map our router's public IP address to an easy-to-remember hostname.

- 1 Point your web browser to <https://www.noip.com/sign-up> and register a new No-IP user account. Ignore or decline any “premium” offers; these DDNS companies make their money by adding features beyond what we need. If you’re reading this while you’re away from your computer, I show you the website in Figure 4.13.
- 2 Part of the signup involves choosing a domain name; as of this writing, [ddns.net](https://ddns.net) is the free option, so I’d suggest you choose that one. As a premium member, you get much more flexibility in your domain names.

You also can choose your own hostname; this is where I plugged in “timwarner.” After all, if my Minecraft friends know me, they know my name and will always remember the name of my Minecraft server!

- 3 After you confirm your account via email and are all logged in, click Hosts/Redirects from the top navigation bar and then Manage Hosts from the side navigation.
- 4 Verify that your custom FQDN matches your router’s public IP address. There’s no magic going on here, by the way. No-IP discerned your public IP address by querying your current default gateway address. Therefore, you might need to manually modify the mapping if you created your account at school but need the mapping to hit your home router.
- 5 If you want, click Dynamic Update Client from the side navigation and download/install this software to your Minecraft server. What it does is check your router’s public IP address every five minutes, and as soon as the client detects a change, it will update your No-IP mapping. Pretty cool, eh?

The screenshot displays the 'Manage Hosts' page in the No-IP user interface. At the top, there's a navigation bar with links like 'Your No-IP', 'Account', 'Support Center', and a 'Sign Out' button. Below this is a secondary navigation bar with 'Hosts / Redirects' selected, along with other services like 'DNS Hosting', 'Domain Registration', etc. On the left side, a sidebar lists various actions like 'Add Host', 'Manage Groups', and 'Download Client'. The main content area is titled 'Manage Hosts' and shows 'Current Hosts: 1'. It features a table with columns for 'Host', 'IP/URL', and 'Action'. The table contains one entry: 'timwarner.ddns.net' with IP '71.228.251.55'. Below the table is a button to 'Add A Host'. At the bottom, there's a section for 'Add Google Apps to your Domain' with a brief description and a 'Learn More' link.

Host	IP/URL	Action
timwarner.ddns.net	71.228.251.55	Modify Remove

**FIGURE 4.13** It’s amazing that companies such as No-IP offer their dynamic DNS services free.

In Figure 4.14 you see that your players can now add your public Minecraft server to their client's server list by using a DNS name instead of a clunky IP address.



**FIGURE 4.14** Your users will appreciate you for advertising your online Minecraft server with a hostname instead of an IP address.

## CAUTION

Please understand that your players might still have to add the `:port` suffix to your hostname if you aren't running your Minecraft server by using the default port address.

## Sweeping Up the Shavings

If you set up a home-based online Minecraft server for experimentation and/or educational purposes, then cool beans. Just be sure to take the proper steps to clean up your environment to maximize your online safety:

- Issue `/stop` in the Minecraft server console to halt the server and prevent it from listening for incoming connection requests.
- Log in to your router and disable your port-forwarding mapping when you no longer need it.
- Delete your No-IP mapping if you aren't using it anymore.



In IT, the security principle of “least service” says that if you don’t need a service running on your network, turn it off and/or uninstall it. This way a malicious user can’t abuse or exploit the service, because it simply isn’t turned on in the first place.

## The Bottom Line

I hope that you feel better about networking than you did before reading this chapter. To be sure, the knowledge and skills you have accrued by now will keep you in excellent stead as we proceed to the next phase of the book.

What is that phase, you ask? Well, it’s time for us to say goodbye to the trusty Mojang “vanilla” Minecraft server and say hello to custom Minecraft servers in general, and the Bukkit platform in particular. See you then!

*This page intentionally left blank*

# INDEX

## A

---

- @a target selector, 151
- Absolute Beginner's Guide to Minecraft Mods Programming* (Cadenhead), 26
- accessing multiplayer game, 6
- accounts, BuyCraft, 220-223
- Adams, Scott, 25
- AdCraft, 217
- addresses
  - connection addresses, 15
  - IP addresses
    - banning*, 62
    - DHCP (Dynamic Host Configuration Protocol)*, 75
    - explained*, 73-74
    - finding*, 39
    - hostnames versus*, 88
    - IPv4 versus IPv6*, 78
    - mapping to hostnames*, 88-90
    - NAT (Network Address Translation)*, 75
    - public versus private addresses*, 74-75
- AdF.ly Skipper extension, 189
- The Advanced Strategy Guide to Minecraft* (O'Brien), 122
- Adventure mode, 6
- Adventurize web portal, 217
- advertising, 228
  - in-game advertising and sponsorship, 216-217
  - on Minecraft server lists, 228-229
  - server registration, 231
  - server status banners, 229-231
  - voting, 232
- Agrarian Skies 2, 241, 245
- agreeing to EULA, 34-35
- Alexander, Scott, 209
- allow-flight property, 50
- allow-nether property, 49
- Anarchy minigame, 14
- Angelou, Maya, 93
- Announcement, 213-214
- announce-player-achievements property, 49
- AntVenom, 243

%appdata% variable, 123

APIs (application programming interfaces)

Bukkit, 172

Minecraft Forge

*explained*, 114

*installing*, 115-116

application programming interfaces.

*See* APIs

Assassin's Creep, 244

Asterion Minecraft, 135

ATLauncher, 134-135

Atonement: Sins of the Past, 241

attacks

DDoS (distributed denial-of-service) attacks, 186-188

DoS (denial-of-service) attacks, 186

grief protection

*blocking grievers*, 190-194

*hacking yourself*, 189

*overview*, 188

*tips for addressing grievers*, 194

auto-afk (Essentials plugin), 107

auto-afk-kick (Essentials plugin), 107

autocomplete, 57

availability of third-party hosts, 162

---

## B

backups, Minecraft Realms

forcing, 155

viewing, 154

/ban command, 61-63, 192

/ban-ip command, 61-63, 192

banners, server status banners, 229-231

basic servers. *See* vanilla servers

BeastNode, 183, 238

Berra, Yogi, 45

BisectHosting, 183, 238

blacklist (Essentials plugin), 108

blocking grievers, 190-194

blocks (command)

*explained*, 150

programming, 152-153

target selectors, 151

BuildCraft, 122

BuildTools instructions, 95

Bukkit, 172

Bukkit

history of, 93-94

plugins. *See* plugins

BukkitWiki, 98

bukkit.yml file, 98

BungeeCord, 208

BuyCraft, 212, 220

creating BuyCraft account, 220-223

creating free packages, 226-227

creating paid packages, 223-225

giving paid item to all players, 225

buying Minecraft license, 7-8, 34

Buy Realms Subscription page  
(Minecraft Realms), 140

## C

---

Cadenhead, Rogers, 26

CaptainSparklez, 243

Capture the Flag minigame, 14

Cauldron

- explained, 113-114

- installing, 117

chat

- chat window, clearing, 57

- console commands for, 55-58

- MCPProHosting, 174

- in multiplayer game, 20-23

chat window, clearing, 57

choosing third-party host, 161-162

Chrome, 100, 189

City minigame, 14

clearing chat window, 57

ClearLagg, 101, 240

client mods, 135

clients

- client mods, 135

- client/server applications, 22

- defined, 22

- Forge client

  - explained*, 114

  - installing*, 115-116

client/server applications, 22

Close-Together Biomes, 245

CloudFlare, 188

cloud services

- advantages/disadvantages, 139

- explained, 137-139

Minecraft Realms

- backups*, 154-155

- creating realms*, 142-144

- explained*, 139-140

- inviting players*, 145-147

- managing players*, 148-150

- restoring worlds*, 157

- subscribing to*, 140-141

- uploading worlds to*, 156-157

Combat Pack, 241

command blocks, 57

- explained, 150

- programming, 152-153

- target selectors, 151

command-costs (Essentials plugin),  
107

console commands

- /ban, 192

- /ban-i, 192

- for chat messages, 55-58

- Essentials plugin, 112

- /gamerule, 148

- /give, 150

- for giving items, 59

- /kick, 192

- /list, 148

- list of, 56

- /mvp, 205

- pl, 120

- for player discipline, 59-64

- /setworldspawn, 148



- in SpigotMC, 96
- /summon, 148
- for teleportation, 59
- /time, 148
- /toggledownfall, 148
- for vanilla servers, 37-42
- config files
  - editing on MCProHosting, 175-176
  - for vanilla servers, 35-37
- Configure Realm screen (Minecraft Realms), 145
- configuring
  - Essentials plugin, 106-108
  - Minecraft server on MCProHosting, 169-170
  - port forwarding, 83-85
  - SpigotMC, 97-98
- config.yml file, 106-108
- connecting multiple worlds, 206-207
- connection addresses, 15
- console commands. *See* commands
- Control Panel (MCProHosting), 168-169
- Cops & Robbers, 14, 244
- CoreProtect, 192
- cosmetic upgrade sales, 217-219
- cracked launchers, avoiding, 8-9
- CraftBukkit, 93-94
- CraftBukkitUpToDate, 240
- Crash Landing, 241
- Create Realm button (Minecraft Realms), 144
- Creative mode, 6
- CreeperHost, 183, 238
- Cube Craft Games, 239
- CubedHost, 238
- Cubeville, 233-234
- currency-symbol (Essentials plugin), 107
- custom-join-message (Essentials plugin), 107
- custom maps, 244-245
- custom-quit-message (Essentials plugin), 107
- custom server entries, creating, 234
- custom servers. *See* plugins
  - Bukkit/CraftBukkit, 93-94
  - Cauldron
    - explained*, 113-114
    - installing*, 117
  - SpigotMC
    - changing version numbers*, 104-105
    - configuring*, 97-98
    - installing*, 95-96
    - reasons for using*, 94
    - starting*, 96-97
    - testing player experience*, 99-100

## D

---

- day/night cycle, 40
- DDNS (Dynamic Domain Name System), 88
- DDoS (distributed denial-of-service) attacks, 186-188

- debug window, opening, 59
- dedicated servers, 161
- Default HD, 242
- default port numbers, 81-83
- demo copy limitations, 7
- denial-of-service (DoS) attacks, 186
- deploying
  - portals, 203-205
  - resource packs with MCProHosting, 177-180
- DHCP (Dynamic Host Configuration Protocol), 75
- Diamond Desert Temple, 245
- difficulty property, 49
- Direwolf20, 241
- Direwolf server, 132
- disciplining players, console commands for, 59-64
- distributed denial-of-service (DDoS) attacks, 186-188
- Diversity 2, 244
- DNS (Domain Name System), 88
- domain name mapping, 180-182
- Domain Name System (DNS), 88
- DonationCraft, 240
- donations, soliciting, 212-216
- DoS (denial-of-service) attacks, 186
- downloading
  - FTB (Feed the Beast), 130
  - JRE (Java Runtime Environment), 27-28
  - PEX (PermissionsEx) plugin, 118

- Dreamcraft, 239
- Dropbox, 177
- Dynamic Domain Name System (DDNS), 88
- Dynamic Host Configuration Protocol (DHCP), 75
- Dyn (hostname service), 88

## E

---

- @e target selector, 151
- Economy, 14, 244
- editing config files on MCProHosting, 175-176
- editions of McMyAdmin, 68
- enable-command-block property, 49
- enable-query property, 49
- enable-rcon property, 50
- end user license agreement (EULA), 11, 210-211
  - agreeing for vanilla servers, 34-35
  - for Minecraft servers, 10
- Engineering with Minecraft (Kelly), 196
- EnviousHost, 238
- Equanimity, 242
- errors. *See* troubleshooting
- Essentials, 101, 240
  - configuring, 106-108
  - installing, 102-105
  - testing player experience, 108-112
- Essentials Wiki, 112

EULA (end user license agreement), 11,  
210-211  
agreeing for vanilla servers, 34-35  
for Minecraft servers, 10  
extensions for Google Chrome, 100

## F

---

Faction, 14, 244  
FadeCloud, 239  
Faithful 32x32, 177, 242  
Feed the Beast (FTB), 14, 241  
downloading, 130  
installing FTB server, 132-133  
setting up, 130-131  
starting FTB launcher, 134  
file extensions, viewing, 31  
finding  
IP addresses, 39  
item IDs, 59  
online servers  
    *connection addresses, 15*  
    *EULA stipulations, 10-11*  
    *Minecraft Forum, 11-14*  
    *public versus whitelist servers, 15-16*  
vanilla servers online, 43-44  
version numbers in SpigotMC, 99-100  
firewalls, 81  
flexibility of third-party hosts, 162  
Floating Island, 245  
force-gamemode property, 49  
forcing Minecraft Realms backups, 155

Forge client  
explained, 114  
installing, 115-116  
FQDN (fully qualified domain name),  
88  
Fragnet, 238  
free packages, creating, 226-227  
FTB (Feed the Beast), 14, 241  
downloading, 130  
installing FTB server, 132-133  
setting up, 130-131  
starting FTB launcher, 134  
fully qualified domain name (FQDN),  
88

## G

---

gamemode property, 49  
/gamerule console command, 40  
/gamerule doDaylightCycle false  
command, 148  
gaming experience, improving, 185  
multiple worlds  
    *connecting, 206-207*  
    *Multiverse Core, 202-203*  
    *Multiverse Portals, 202*  
    *portal deployment, 203-205*  
    *teleporting between, 203*  
server protection  
    *DDoS (distributed denial-of-service)*  
    *attacks, 186-188*  
    *grief protection, 188-194*  
    *motivations of server abusers, 186*

- world spawn, 194
  - building spawn lobbies, 196-198*
  - setting server spawn, 195*
  - WorldEdit, 199-201*
- Gandhi, Mahatma, 185
- generate-structures property, 50
- generator-settings property, 49
- generic servers. *See* vanilla servers
- Get Realms button (Minecraft Realms), 140
- GGServers, 183, 238
- GigaTech, 135
- /give command, 59, 150
- GiveAllItems Bukkit plugin, 225
- giving items
  - console commands for, 59
  - giving paid items to all players, 225
- GoDaddy, 180-182
- Google Chrome, 100, 189
- griefing, 15, 59-60
- grief protection
  - blocking griefers, 190-194
  - hacking yourself, 189
  - overview, 188
  - tips for addressing griefers, 194
- GroupManager, 240

## H

---

- hacking yourself, 189
- hacktivism, 186
- Hardcore mode, 6

- hardcore property, 49
- Hardcore PvP minigame, 14
- /help console command, 19-20, 37
- help system in multiplayer game, 19-20
- Herobrine's Mansion, 244
- Hexxit, 241
- history of Bukkit/CraftBukkit, 93-94
- hostnames, 88
  - DNS (Domain Name System), 88
  - No-IP configuration, 88-90
- hosts, top 10 Minecraft hosts, 237-238
- Hotel Paradise, 245
- Hunger Games minigame, 14
- Hypixel, 239

## I

---

- identity theft, 186
- improving gaming experience, 185
  - multiple worlds
    - connecting, 206-207*
    - Multiverse Core, 202-203*
    - Multiverse Portals, 202*
    - portal deployment, 203-205*
    - teleporting between, 203*
- server protection
  - DDoS (distributed denial-of-service) attacks, 186-188*
  - grief protection, 188-194*
  - motivations of server abusers, 186*

- world spawn, 194
  - building spawn lobbies, 196-198*
  - setting server spawn, 195*
  - WorldEdit, 199-201*
- IndustrialCraft2, 122
- in-game advertising and sponsorship, 216-217
- installing
  - Cauldron, 117
  - Forge client, 115-116
  - FTB (Feed the Beast) server, 132-133
  - JRE (Java Runtime Environment), 27-28
  - McMyAdmin, 66
  - Minecraft, 8
  - mods, 122-123
  - plugins
    - Essentials plugin, 102-105*
    - MCPProHosting, 172-173*
    - number to install, 102*
  - Railcraft, 122-123
  - SpigotMC, 95-96
  - vanilla servers
    - on Mac OS X, 31-33*
    - on Windows, 29-30*
- Internet Protocol (IP) addresses. *See* IP addresses
- Intrepid, 186
- inviting players in Minecraft Realms, 145-147
- IP addresses
  - banning, 62
  - DHCP (Dynamic Host Configuration Protocol), 75

- explained, 73-74
- finding, 39
- hostnames versus, 88
- IPv4 versus IPv6, 78
- mapping to hostnames, 88-90
- NAT (Network Address Translation), 75
- public versus private addresses, 74-75
- IPv4 addresses, 78
- IPv6 addresses, 78
- item IDs, finding, 59
- item-spawn-blacklist (Essentials plugin), 107

## J

---

- Java environment preparation, 25
  - JRE installation, 27-28
  - verifying Java version, 26-27
- Java Runtime Environment (JRE), 27-28
- JavaScript Object Notation (JSON), 37
- java - version command errors, 27
- JeromeASF, 243
- John Smith Legacy, 242
- joining
  - MCPProHosting, 163-168
  - online servers, 16-19
- JRE (Java Runtime Environment), 27-28
- JSON (JavaScript Object Notation), 37

## K

---

Kelly, James Floyd, 196  
/kick command, 60-61, 192  
kit:tools (Essentials plugin), 108  
kits (Essentials plugin), 107

## L

---

LAN (local area network) servers  
    defined, 10  
    opening single-player game to, 68-70  
    user connections, 50-52  
launchers  
    Asterion Minecraft, 135  
    ATLauncher, 134-135  
    avoiding cracked, 8-9  
    FTB (Feed the Beast) launcher, 134  
    GigaTech, 135  
    TechnicPack, 135  
    VoidLauncher, 135  
level-name property, 50  
level-seed property, 50  
level-type property, 49  
license for Minecraft, buying, 7-8, 34  
Lichcraft, 239  
/list command, 21, 148  
lists, advertising on, 228-229  
lobbies (spawn), building, 196-198  
local area network (LAN) servers  
    defined, 10  
    opening single-player game to, 68-70  
    user connections, 50-52

location coordinates, 59  
logfiles, reading, 52-54  
logging in to MCProHosting, 170-171

## M

---

Mac OS X  
    finding IP addresses, 39  
    startup scripts, 96  
    TCP/IP configuration settings, 77  
    vanilla server installation, 31-33  
    verifying Java version, 27  
    viewing file extensions, 31  
Magic Farm 3: Harvest, 241  
managing  
    players in Minecraft Realms, 148-150  
    vanilla servers with McMyAdmin,  
        65-68  
Manneh, Carl, 137, 157  
mapping  
    hostnames to IP addresses, 88-90  
    servers to domains, 180-182  
maps, top 10 Minecraft custom maps,  
    244-245  
MarioKart, 242  
max-build-height property, 50  
max-players property, 49  
max-tick-time property, 49  
max-world-size property, 49  
MCEdit, 197-198  
MC-Legends, 239  
McMyAdmin, 65-68

- MCProHosting, 238
  - chat, 174
  - configuring and starting Minecraft server, 169-170
  - Control Panel, 168-169
  - deploying resource packs, 177-180
  - domain name mapping, 180-182
  - editing config files, 175-176
  - installing plugins, 172-173
  - joining, 163-168
  - logging in, 170-171
- MCSchematics, 197
- /me console command, 58
- Mega Mountains, 246
- message of the day (MOTD), 235
- Micro Cubes, 245
- Microsoft's purchase of Mojang, 211
- Minecraft EULA (end user license agreement), 210-211
- Minecraft Forge
  - explained, 114
  - installing, 115-116
- Minecraft Forum, 11-14, 184
- Minecraft hosts, 237-238
- Minecraftertopia, 59
- Minecraft Realms, 10
  - backups
    - forcing*, 155
    - viewing*, 154
  - creating realms, 142-144
  - explained, 139-140
  - inviting players, 145-147
  - managing players, 148-150
  - restoring worlds, 157
  - subscribing to, 140-141
  - uploading worlds to, 156-157
- Minecraft Schematics, 197
- Minecraft seeds, 245-246
- Minecraft server lists, advertising on, 228-229
- Minecraft Wiki, 56
- Mineplex, 239
- Minevolution, 245
- minigames, 14-15, 243-244
- modpacks
  - BuildCraft, 122
  - FTB (Feed the Beast), 14, 241
    - downloading*, 130
    - installing FTB server*, 132-133
    - setting up*, 130-131
    - starting FTB launcher*, 134
  - IndustrialCraft2, 122
  - launchers, 134-135
  - Pixelmon, 122
  - Railcraft
    - explained*, 122
    - installing*, 122-123
    - testing*, 125-129
  - Tinkers Construct, 122
  - top 10 modpacks, 240-241
- mods, 113, 122
  - BuildCraft, 122
  - Cauldron project, 113-114
  - client mods, 135

- FTB (Feed the Beast), 14, 241
  - downloading*, 130
  - installing FTB server*, 132-133
  - setting up*, 130-131
  - starting FTB launcher*, 134
- IndustrialCraft2, 122
- launchers, 134-135
- Minecraft Forge
  - explained*, 114
  - installing*, 115-116
- Optifine, 135
- Pixelmon, 122
- plugins versus, 100
- programming, 26
- Railcraft
  - explained*, 122
  - installing*, 122-123
  - testing*, 125-129
- Rei's Minimap, 135
- Tekkit, 122
- Tinkers Construct, 122
- TooManyItems, 135
- monetizing Minecraft server
  - advertising your server
    - on Minecraft server lists*, 228-229
    - server registration*, 231
    - server status banners*, 229-231
    - voting*, 232
- BuyCraft
  - creating BuyCraft account*, 220-223
  - creating free packages*, 226-227
  - creating paid packages*, 223-225
  - giving paid item to all players*, 225
  - cosmetic upgrade sales, 217-219
  - Cubeville example, 233-234
  - custom server entries, 234
  - donations, 212-216
  - in-game advertising and sponsorship, 216-217
  - Minecraft EULA (end user license agreement), 210-211
  - Mojang's position on, 210
  - MOTD (message of the day), 235
  - overview, 209
  - VIP server access/subscriptions, 219-220
- MOTD (message of the day), 235
- motd property, 50
- motivations of server abusers, 186
- Multicraft, 162
- multiplayer game
  - accessing, 6
  - chatting in, 20-23
  - defined, 10
  - game types, 10
  - help system, 19-20
  - minigames, 14-15
  - netiquette, 19, 23-24
  - opening single-player game as, 68-70
- multiple server instances, running, 42-43
- multiple worlds
  - connecting, 206-207
  - Multiverse Core, 202-203
  - Multiverse Portals, 202



- portal deployment, 203-205
- teleporting between, 203
- Multiverse, 240
- Multiverse Core, 102, 202-203
- Multiverse Portals, 202
- /mvp command, 205
- MythCraft, 239

## N

---

- NAT (Network Address Translation), 75
- Nether Fortress, 246
- netiquette in multiplayer game, 19, 23-24
- Network Address Translation (NAT), 75
- network-compression-threshold property, 49
- networks
  - DHCP (Dynamic Host Configuration Protocol), 75
  - firewalls, 81
  - hostnames
    - DNS (*Domain Name System*), 88
    - No-IP configuration, 88-90
  - IP addresses
    - banning, 62
    - DHCP (*Dynamic Host Configuration Protocol*), 75
    - explained, 73-74
    - finding, 39
    - hostnames versus, 88
    - IPv4 versus IPv6, 78
    - mapping to hostnames, 88-90
    - NAT (*Network Address Translation*), 75
    - public versus private addresses, 74-75
  - NAT (Network Address Translation), 75
  - port forwarding
    - configuring, 83-85
    - explained, 81-83
    - testing, 85-87
  - port numbers, 81-83
  - router configuration settings, 78-80
  - security issues, 90-91
  - TCP/IP configuration settings, 76-78
  - terminology, 73
  - topology, 72
- Nitrous Networks, 238
- NoCheatPlus, 190
- NodeCraft, 238
- Nodus, 188
- No-IP (hostname service), 88-90

## O

---

- O'Brien, Stephen, 122
- online-mode property, 50
- online servers
  - defined, 10
  - finding
    - connection addresses, 15
    - EULA stipulations, 10-11
    - Minecraft Forum, 11-14
    - public versus whitelist servers, 15-16
    - vanilla servers, 43-44
  - joining, 16-19

- on-premises services, 137
- /op console command, 38
- opening
  - debug window, 59
  - single-player game as multiplayer, 68-70
- op-permission-level property, 49
- OPs, 22
- ops-name-color (Essentials plugin), 107
- Optifine, 135
- OS X. *See* Mac OS X
- Ownage Hosting, 183

## P

---

- @p target selector, 151
- packages, creating
  - free packages, 226-227
  - paid packages, 223-225
- paid items, giving to all players, 225
- paid packages, creating, 223-225
- /pardon console command, 63-64
- /pardon-ip console command, 63-64
- Parkour, 14, 243
- passwords, 80
- Perlis, Alan, 71
- PermissionsEx (PEX) plugin, 240
  - downloading, 118
  - quick start, 118-120
  - testing, 120-121
- PermissionsEx plugin, 101
- PEX (PermissionsEx) plugin, 240
  - downloading, 118
  - quick start, 118-120
  - testing, 120-121
- Pixelmon, 122, 241
- PixelmonCraft, 239
- Pixel Perfection, 242
- player connections to vanilla servers, 86-87
- player experience, testing
  - Essentials plugin, 108-112
  - in SpigotMC, 99-100
- player-idle-timeout property, 49
- player interactions
  - chat messages, 55-58
  - disciplining players, 59-64
  - giving items, 59
  - inviting in Minecraft Realms, 145-147
  - managing in Minecraft Realms, 148-150
  - teleportation, 59
- player list, viewing, 56
- Player versus Environment (PvE) minigame, 15
- Player versus Player (PvP) minigame, 14
- pl command, 120
- plugins
  - AdCraft, 217
  - Announcement, 213-214
  - BuyCraft. *See* BuyCraft
  - configuring, 106-108
  - defined, 100
  - GiveAllItems, 225

- installing, 102-105
- MCPProHosting, 172-173
- mods versus, 100
- most popular, 100-102
- NoCheatPlus, 190
- number to install, 102
- PEX (PermissionsEx)
  - downloading*, 118
  - quick start*, 118-120
  - testing*, 120-121
- testing player experience, 108-112
- top 10 Bukkit plugins, 240
- VIP, 219-220
- Popular MMOs, 243
- portals, deploying, 203-205
- PortCheckTool.com, 85
- port forwarding
  - configuring, 83-85
  - explained, 81-83
  - testing, 85-87
- port numbers, 81-83
- prevent:spawn (Essentials plugin), 107
- Prison, 15, 244
- privacy issues, 76
- private IP addresses, 74-75
- programming mods, 26
- programming command blocks, 152-153
- protecting servers, 186
  - DDoS (distributed denial-of-service) attacks, 186-188
  - grief protection
    - blocking grievers*, 190-194
    - hacking yourself*, 189
    - overview*, 188
    - tips for addressing grievers*, 194
  - motivations of server abusers, 186
- proxies, 208
- public IP addresses, 74-75
- public servers
  - top 10 servers, 239
  - whitelist servers versus, 15-16
- purchasing Minecraft license, 7-8, 34
- PvP, 244
- pvp property, 49

## Q-R

---

- @r target selector, 151
- Railcraft
  - explained, 122
  - installing, 122-123
  - testing, 125-129
- rain, starting/stopping, 40
- RAM (random access memory) system
  - requirements, 34
- reading logfiles, 52-54
- Realms, 10
  - backups
    - forcing*, 155
    - viewing*, 154
  - creating realms, 142-144
  - explained, 139-140
  - inviting players, 145-147

- managing players, 148-150
- restoring worlds, 157
- subscribing to, 140-141
- uploading worlds to, 156-157
- Realm Templates screen (Minecraft Realms), 144
- Reddit MinecraftSchematics, 197
- registering servers, 231
- Rei's Minimap, 135
- repositories
  - Bukkit plugins, 100
  - Essentials plugin, 102
- reputation of third-party hosts, 161
- resource-pack-hash property, 49
- resource-pack property, 50
- resource packs
  - deploying with MCProHosting, 177-180
  - top 10 resource packs, 242
- restoring worlds in Minecraft Realms, 157
- router configuration settings, 78-80
- routers, port forwarding
  - configuring, 83-85
  - testing, 85-87
- Runescape Pack, 242
- running multiple server instances, 42-43
- Russell, Bertrand, 113

## S

---

- /say console command, 40, 55
- schematics, 197-198
  - popular schematic sites, 197
  - preparing land for, 197
    - with MCEdit*, 197-198
    - with WorldEdit*, 199-201
- security, 186
  - DDoS (distributed denial-of-service) attacks, 186-188
  - grief protection
    - blocking grievers*, 190-194
    - hacking yourself*, 189
    - overview*, 188
    - tips for addressing grievers*, 194
  - motivations of server abusers, 186
  - network services, 90-91
  - routers, 80
  - vanilla servers, warning about, 76
- seeds, 245-246
- Select Template button (Minecraft Realms), 144
- server console, 53
- server-ip property, 50
- server monetization
  - advertising your server
    - on Minecraft server lists*, 228-229
    - server registration*, 231
    - server status banners*, 229-231
    - voting*, 232
- BuyCraft
  - creating BuyCraft account*, 220-223
  - creating free packages*, 226-227
  - creating paid packages*, 223-225
  - giving paid item to all players*, 225
- cosmetic upgrade sales, 217-219
- Cubeville example, 233-234

- custom server entries, 234
- donations, 212-216
- in-game advertising and sponsorship, 216-217
- Minecraft EULA (end user license agreement), 210-211
- Mojang's position on, 210
- MOTD (message of the day), 235
- overview, 209
- VIP server access/subscriptions, 219-220
- server-port property, 50
- server.properties file
  - configuring SpigotMC, 98
  - settings, 48-50
- server registration, 231
- servers
  - custom servers
    - Bukkit/CraftBukkit history*, 93-94
    - Cauldron*, 113-114, 117
    - changing version numbers*, 104-105
    - installing SpigotMC*, 95-100
  - dedicated servers, 161
  - defined, 22
  - Direwolf, 132
  - FTB (Feed the Beast) server, installing, 132-133
  - LAN servers
    - defined*, 10
    - opening single-player game to*, 68-70
  - mapping to domains, 180-182
  - Minecraft Realms, 10
  - monetization
    - advertising your server*, 228-232
    - BuyCraft*, 220-227
    - cosmetic upgrade sales*, 217-219
    - Cubeville example*, 233-234
    - custom server entries*, 234
    - donations*, 212-216
    - in-game advertising and sponsorship*, 216-217
    - Minecraft EULA (end user license agreement)*, 210-211
    - Mojang's position on*, 210
    - MOTD (message of the day)*, 235
    - overview*, 209
    - VIP server access/subscriptions*, 219-220
  - online servers
    - connection addresses*, 15
    - defined*, 10
    - EULA stipulations*, 10-11
    - finding*, 11-14
    - joining*, 16-19
    - public versus whitelist servers*, 15-16
  - protecting
    - DDoS (distributed denial-of-service) attacks*, 186-188
    - grief protection*, 188-194
    - motivations of server abusers*, 186
  - registering, 231
  - system requirements, 34
  - top 10 public Minecraft servers, 239
  - uptimes, 13
  - vanilla servers
    - advantages/disadvantages*, 28-29
    - agreeing to EULA*, 34-35

- chat messages*, 55-58
- configuration files*, 35-37
- console commands*, 37-42
- finding online*, 43-44
- giving items*, 59
- installing on Mac OS X*, 31-33
- installing on Windows*, 29-30
- Java environment preparation*, 25-28
- logfiles*, 53-54
- managing with McMyAdmin*, 65-68
- mapping IP address to hostname*, 88-90
- meaning of term*, 28, 34
- player connections*, 86-87
- player discipline*, 59-64
- port forwarding configuration*, 83-85
- running multiple instances*, 42-43
- security*, 90-91
- security issues*, 76
- server console UI*, 53
- server.properties file settings*, 48-50
- teleportation*, 59
- test environment*, 45-46
- testing port forwarding*, 85-87
- user connections via LAN*, 50-52
- UUIDs*, 53-55
- VPSs (virtual private servers), 160
- server specifications (third-party hosts), 161
- server status banners, 229-231
- services
  - cloud services. *See also* Minecraft Realms
    - advantages/disadvantages*, 139
    - explained*, 137-139
  - on-premises services, 137
- third-party hosting
  - BeastNode*, 183
  - Bisect Hosting*, 183
  - choosing hosts*, 161-162
  - Creeper Host*, 183
  - dedicated servers*, 161
  - GGServers*, 183
  - MCProHosting*, 163-182
  - overview*, 159
  - Ownage Hosting*, 183
  - VPSs (virtual private servers)*, 160
- /setworldspawn command, 148
- showing. *See* viewing
- signing up with Minecraft Realms, 140-141
- signs (Essentials plugin), 107
- single-player game
  - modes of play, 6
  - opening to multiplayer, 68-70
- Skyblock, 15, 243
- SkyDoesMinecraft, 243
- Skywars, 244
- snooper-enabled property, 49
- Soartex Fanver, 242
- Soartex Invictus, 242
- soliciting donations, 212-216
- spawn-animals property, 50
- Spawn Beside Jungle Temple, 245
- spawn lobbies, building, 196-198
- spawn-monsters property, 49
- spawn-npcs property, 50

- spawn-protection property, 48
- Spectator mode, 6
- SpigotMC
  - changing version numbers, 104-105
  - configuring, 97-98
  - installing, 95-96
  - reasons for using, 94
  - starting, 96-97
  - testing player experience, 99-100
- SpigotMC Wiki, 98
- spigot.yml file, 98
- Spleef, 244
- sponsorship, 216-217
- SSundee, 243
- Stampy, 243
- Stampy Cat, 209
- stampylonghead YouTube channel, 209
- Stampy World, 245
- starting
  - FTB (Feed the Beast) launcher, 134
  - Minecraft server on MCProHosting, 169-170
  - SpigotMC, 96-97
- starting-balance (Essentials plugin), 107
- startup scripts, creating, 96
- status banners, 229-231
- /stop console command, 40-42
- Stronghold in Ravine, 245
- Sublime Text, 47-48
- subnet masks, 73-74
- subscribing to Minecraft Realms, 140-141

- /summon command, 148
- Sun Tzu, 5
- Survival Games, 244
- Survival mode, 6
- Swamp and Witch Hut, 246
- The Syndicate Project, 243
- system requirements
  - for McMyAdmin, 67
  - for Minecraft, 7
  - for servers, 34

## T

---

- Tab key
  - autocomplete, 57
  - viewing player list, 56
- target selectors, 151
- TCP/IP (Transmission Control Protocol/Internet Protocol), 73, 76-78
- TechnicPack, 135
- Tekkit, 15, 122, 241
- teleportation
  - console commands for, 59
  - teleporting between worlds, 203
- teleport-cooldown (Essentials plugin), 107
- /tell console command, 20, 57
- terminology for networks, 73
- testing
  - Essentials plugin, 108-112
  - PEX (PermissionsEx) plugin, 120-121

- player experience in SpigotMC, 99-100
- port forwarding, 85-87
- Railcraft, 125-129
- test environment for vanilla servers, 45-46
- text editors, Sublime Text, 47-48
- texture packs, 242
- TheArchon, 239
- TheBajanCanadian, 243
- TheDiamondMinecart, 243
- third-party hosting
  - BeastNode, 183
  - Bisect Hosting, 183
  - choosing hosts, 161-162
  - Creeper Host, 183
  - dedicated servers, 161
  - GGServers, 183
  - MCPProHosting
    - chat*, 174
    - configuring and starting Minecraft server*, 169-170
    - Control Panel*, 168-169
    - deploying resource packs*, 177-180
    - domain name mapping*, 180-182
    - editing config files*, 175-176
    - installing plugins*, 172-173
    - joining*, 163-168
    - logging in*, 170-171
  - overview, 159
  - Ownage Hosting, 183
  - VPSs (virtual private servers), 160
- /time console command, 40
- /time set day 1000 command, 148
- Tinkers Construct, 122
- /toggledownfall command, 40, 148
- TolkienCraft II, 244
- TooManyItems, 135
- Top 10 lists, 237
  - Bukkit plugins, 240
  - custom maps, 244-245
  - Minecraft hosts, 237-238
  - Minecraft seeds, 245-246
  - Minecraft YouTubers, 243
  - minigames, 243-244
  - modpacks, 240-241
  - public Minecraft servers, 239
  - resource packs, 242
- topology of networks, 72
- TownCraft, 194
- /tp console command, 59
- Transmission Control Protocol/Internet Protocol (TCP/IP), 73
- troubleshooting
  - java - version command errors, 27
  - port forwarding errors, 86
  - SpigotMC console commands, 96
  - typos, 33
- typos, 33

---

## U

- UI (user interface) for server console, 53
- Underwater Temple, 245
- universally unique identifiers (UUIDs), 53-55



- uploading worlds to Minecraft Realms, 156-157
- uptimes for servers, 13
- user connections via LAN, 50-52
- user interface (UI) for server console, 53
- UUIDs (universally unique identifiers), 53-55

## V

---

### vanilla servers

- advantages/disadvantages, 28-29
- agreeing to EULA, 34-35
- chat messages, console commands for, 55-58
- configuration files, 35-37
- console commands, 37-42
- finding online, 43-44
- giving items, console commands for, 59
- installing
  - on Mac OS X, 31-33
  - on Windows, 29-30
- Java environment preparation, 25
  - JRE installation, 27-28
  - verifying Java version, 26-27
- logfiles, 53-54
- managing with McMyAdmin, 65-68
- mapping IP address to hostname, 88-90
- meaning of term, 28, 34
- player connections, 86-87

- player discipline, console commands for, 59-64
- port forwarding configuration, 83-85
- running multiple instances, 42-43
- security, 90-91
- security issues, 76
- server console UI, 53
- server.properties file settings, 48-50
- teleportation, console commands for, 59
- test environment, 45-46
- testing port forwarding, 85-87
- user connections via LAN, 50-52
- UUIDs, 53-55
- verifying Java version, 26-27
- version numbers (SpigotMC)
  - finding, 99-100
  - changing, 104-105
- view-distance property, 50
- viewing
  - file extensions, 31
  - Minecraft Realms backups, 154
  - player list, 56
  - router configuration settings, 78-80
  - TCP/IP configuration settings, 76-78
- Village with Lots of Loot, 245
- VIP plugin, 219-220
- VIP server access/subscriptions, 219-220
- virtual private servers (VPSs), 160
- VoidLauncher, 135
- Votifier, 240
- voting, 232
- VPSs (virtual private servers), 160

## W

---

weather, starting/stopping rain, 40

WesterosCraft, 244

white-list property, 50

/whitelist console command, 63-64

whitelist servers, 15-16

Windows

- finding IP addresses, 39

- java - version command errors, 27

- McMyAdmin installation, 66

- startup scripts, 96

- TCP/IP configuration settings, 76

- vanilla server installation, 29-30

- verifying Java version, 26

- viewing file extensions, 31

WorldEdit, 102, 197-201, 240

WorldGuard, 101, 240

worlds

- restoring in Minecraft Realms, 157

- uploading to Minecraft Realms, 156-157

world spawn, 194

- building spawn lobbies, 196-198

- setting server spawn, 195

- WorldEdit, 199-201

Wurst, 188

## X-Y-Z

---

Xray Ultimate, 242

YAML file format, 98

YouTubers, top 10 Minecraft

- YouTubers, 243