



5TH EDITION

NGINX HTTP Server

Harness the power of NGINX with a series of detailed tutorials and real-life examples

GABRIEL OUIRAN | CLEMENT NEDELCU
MARTIN BJERRETOFT FJORDVALD

NGINX HTTP Server

Harness the power of NGINX with a series of detailed tutorials and real-life examples

Gabriel Ouiran

Clement Nedelcu

Martin Bjerretoft Fjordvald



NGINX HTTP Server

Copyright © 2024 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

The author acknowledges the use of cutting-edge AI, such as ChatGPT, with the sole aim of enhancing the language and clarity within the book, thereby ensuring a smooth reading experience for readers. It's important to note that the content itself has been crafted by the author and edited by a professional publishing team.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the authors, nor Packt Publishing or its dealers and distributors, will be held liable for any damages caused or alleged to have been caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

Group Product Manager: Dhruv J. Kataria

Publishing Product Manager: Khushboo Samkaria

Book Project Manager: Srinidhi Ram

Senior Editor: Adrija Mitra

Technical Editor: Nithik Cheruvakodan

Copy Editor: Safis Editing

Proofreader: Safis Editing

Indexer: Hemangini Bari

Production Designer: Shankar Kalbhor

DevRel Marketing Coordinator: Shruthi Shetty

Senior DevRel Marketing Executive: Marylou De Mello

First published: July 2010

Second edition: July 2013

Third edition: November 2015

Fourth edition: February 2018

Fifth edition: June 2024

Production reference: 1080524

Published by Packt Publishing Ltd.

Grosvenor House

11 St Paul's Square

Birmingham

B3 1RB, UK

ISBN 978-1-83546-987-3

www.packtpub.com

To the incredible circle of friends and mentors who've guided and helped me, to the community that has always inspired me, and to all those who hold a special place in my heart, filling my life with joy and purpose – you truly are my pillars of strength.

– Gabriel Ouiran

Contributors

About the authors

Gabriel Ouiran is a systems administrator at Edeka Software, Singapore, specializing in IT infrastructure with a strong focus on AI and cloud technologies. Gabriel is also a passionate tech enthusiast and a free and open source software contributor. He has over 10 years of experience working in IT for different companies. Gabriel now applies this expertise to enhance educational software solutions.

I'm grateful to Clément for the chance to write this fifth edition. Special thanks to Remy, Jean-Vincent, and Inès for their invaluable support and assistance in writing this book.

Clement Nedelcu was born in France and studied at UK, French, and Chinese universities. After teaching computer science, programming, and systems administration at several eastern Chinese universities, he worked as a technology consultant in France, specialized in the web and .NET software development as well as Linux server administration. Since 2005, he has also been administering a major network of websites in his spare time, which allowed him to discover NGINX. Clement now works as CTO. for a Singapore-based company developing management software for schools.

Martin Bjerretoft Fjordvald is a 35-year-old Danish entrepreneur who started his company straight out of high school. Backed by a popular website, he became a jack of all trades having to deal with the business, programming, and marketing side of his business. The popularity of his website grew and so did the performance requirements of his code and servers.

He got involved with the community project to document NGINX early on and has written several blog posts and wiki articles detailing how NGINX works.

About the reviewers

Rémy Beaufile has worked in the field of IT security for the last 10 years. He has performed security audit and consulting for both the private and public sector, from small local companies to international conglomerates. His current job was as security engineer for a French Insurtech start-up named Luko.

Rémy is also a contributor to several open source projects linked to automation and social media, and he is an active participant in several security-oriented communities.

I believe that no matter how small the participation, all of us can build a better digital future in the open source space, far from the conflicts of interest moving the private sector.

I'd also like to thank Ronnie, my cat, who helped me review and test all chapters of this book.

Jean-Vincent HAY has been a security auditor and DevSecOps consultant for more than eight years and currently works for Squad in France. His main areas of interest are application security, especially on mobile applications and their backends, and he mainly works for the industrial and banking sectors.

I hope this book helps people do a better job at giving me less work in raising awareness about OWASP and the whole security auditing industry. I'd like to thank Squad for giving me so many occasions to learn about good and bad NGINX configuration, and a community of pirate octopuses for their heads-up. Of course, this would have been much more difficult without the love of my life.

Table of Contents

Part 1: Begin with NGINX

1

Downloading and Installing NGINX 3

Installing NGINX via package managers	4	Path options	12
NGINX-provided packages	4	Build configuration issues	13
Compiling NGINX from source	5	Compiling and installing	14
Installing GNU Compiler Collection	6	Controlling the NGINX service	15
The PCRE library	7	Daemons and services	15
The zlib library	7	Users and groups	15
OpenSSL	8	NGINX command-line switches	16
Downloading and compiling the NGINX source code	8	Starting and stopping the daemon	16
Websites and resources	9	Testing the configuration	17
Version branches	10	Other switches	18
Features	10	Adding NGINX as a system service	18
Downloading and extracting	11	systemd unit file	18
Exploring the options for configuring the compilation	11	Handling system errors	20
The easy way	12	A quick overview of the possibilities offered by NGINX Plus	20
		Summary	21

2

Basic NGINX Configuration 23

Delving into the configuration file syntax	23	Core module directives	32
Configuration directives	24	The events module	38
Organization and inclusions	26	Configuration module	40
Directive blocks	27	Necessary adjustments	40
Advanced language rules	29	Testing your server	41
Looking at the base module directives	31	Creating a test server	41
What are base modules?	31	Upgrading NGINX gracefully	43
NGINX process architecture	31	Summary	43

Part 2: Dive into NGINX

3

Exploring the HTTP Configuration 47

An introduction to the HTTP core module and its three new blocks	47	http2_max_header_size	75
Exploring the HTTP core module directives	49	http2_max_requests	75
Socket and host configuration	50	http2_recv_buffer_size	75
Paths and documents	53	http2_recv_timeout	75
Client requests	57	Module variables	76
MIME types	62	Exploring the variables introduced by the HTTP core module	76
Limits and restrictions	64	Request headers	76
File processing and caching	67	Response headers	77
Other directives	69	NGINX-generated headers	78
Exploring the directives of HTTP/2	73	Understanding and exploring the location block	80
http2_chunk_size	74	Location modifier	80
http2_body_preload_size	74	Search order and priority	83
http2_idle_timeout	74	Summary	86
http2_max_concurrent_streams	74		
http2_max_field_size	74		

4

Exploring Module Configuration in NGINX 87

Exploring the Rewrite module	87	Website access and logging	105
Reminder of regular expressions	88	Limits and restrictions	109
Internal requests	93	Content and encoding	113
Conditional structure	98	About your visitors	125
Directives	100	SSL and security	131
Common rewrite rules	103	Other miscellaneous and optional modules	136
Looking at some additional modules	105	Summary	138

5

PHP and Python with NGINX 139

Introduction to FastCGI technologies	139	PHP-FPM	152
Understanding the CGI mechanism	140	Setting up PHP and PHP-FPM	152
CGI	141	NGINX configuration	154
FastCGI	142	Python and NGINX	155
uWSGI and SCGI	143	Django	155
Main directives	144	Setting up Python and Django	156
FastCGI caching and buffering	150	NGINX configuration	157
PHP with NGINX	151	Summary	158
Architecture	152		

6

NGINX as a Reverse Proxy 159

Exploring the reverse proxy mechanism	159	Other directives	170
Exploring the NGINX proxy module	160	Variables	172
Main directives	161	Looking at NGINX and microservices	172
Caching, buffering, and temporary files	164	Summary	173
Limits, timeouts, and errors	168		

Part 3: NGINX in Action

7

Introduction to Load Balancing and Optimization 177

Introducing load balancing	177	The stream module	184
Understanding the concept of load balancing	178	An example of MySQL load balancing	184
Session affinity	179	Exploring thread pools and I/O mechanisms	185
The upstream module	180	Relieving worker processes	185
Request distribution mechanisms	182	AIO, Sendfile, and DirectIO	186
Using NGINX as a TCP/UDP load balancer	183	Summary	187

8

NGINX within a Cloud Infrastructure 189

Understanding cloud infrastructure	190	Setting up NGINX inside Docker	194
The traditional approach	190	Integrating PHP with NGINX using Docker	
The cloud approach	190	Compose	195
Using Docker	192	Setting up NGINX inside Docker to proxy host applications	197
Installing Docker	192	Summary	198
Your first Docker container	192		
Simplifying with Docker Compose	193		

9

Fully Deploy, Manage, and Auto-Update NGINX with Ansible 199

Understanding configuration management	199	Setting up automatic updates using Ansible	203
Running your first Ansible playbook	200	Summary	204
Setting up NGINX using Ansible	202		

10

Case Studies 205

Exploring SSL Certificates and HTTPS by default	206	Preparing your server and obtaining WordPress	209
Certificate Management with acme.sh	206	NGINX configuration	212
acme.sh and the DNS API	207	WordPress configuration	215
Issuing a signed certificate	208	Deploying Nextcloud	217
Centralizing SSL Configuration with NGINX	208	Getting Nextcloud	217
Implementing HTTP/2 with SSL	209	Summary	219
Deploying a WordPress site	209		

11

Troubleshooting 221

Looking at some general tips on NGINX troubleshooting	221	Exploring 400 Bad Request	227
Checking access permissions	222	Looking at truncated or invalid FastCGI responses	227
Testing your configuration	222	Exploring location block priorities	228
Have you reloaded the service?	222	Looking at if block issues	228
Checking logs	223	Inefficient statements	228
Installing a log parser	223	Unexpected behavior	229
Troubleshooting install issues	225	Summary	230
Looking at the 403 forbidden custom error page	226		

Index 231

Other Books You May Enjoy 240

Preface

NGINX, known for its lightweight design, is a powerhouse HTTP server designed for handling high-traffic websites, with network scalability as its primary objective. In our increasingly connected world, optimizing your web apps has become more critical than ever. Whether you are a beginner or an experienced administrator, this NGINX book will guide you through the complete process of setting up this lightweight HTTP server, from a quick and basic configuration to a more detailed configuration tailored to your needs. This new edition focuses on the latest version 1.25.2, which introduces new features such as HTTP/3 and QUIC.

Packed with a multitude of real-world examples, this book will help you to secure your infrastructure with automatic TLS certificates, expertly place NGINX in front of your existing applications, and do much more. You'll also cover orchestration, Docker, bandwidth management, OpenResty, and NGINX Plus commercial features to enhance and optimize your infrastructure or design a brand-new architecture.

By the end of this book, you'll be able to adapt and use a wide variety of NGINX implementations to solve any problems you have.

Who this book is for

This edition of our NGINX book is meticulously designed for beginner DevOps engineers, system administrators, and web developers who want to improve their understanding of web server management, especially in the areas of performance optimization and cloud infrastructure. Whether you're just beginning your journey into web technologies or looking to consolidate your base, this book offers practical insights and hands-on experience with NGINX, the heart of modern web applications.

The primary personas who will find this content invaluable are beginner DevOps engineers, system administrators, and curious web developers. DevOps newcomers will learn NGINX's essentials, from setup to cloud optimization, while system administrators will gain strategies for effective deployment and troubleshooting, and developers will discover how to enhance their projects with improved performance, security, and scalability.

What this book covers

Chapter 1, Downloading and Installing NGINX, is an introduction to acquiring and configuring NGINX on your system. It covers the essentials, from the initial download to ensuring NGINX persists across reboots by installing it as a system service. Key areas include preparing your system, configuring NGINX, managing dynamically loaded modules, and integrating NGINX as a reliable system service. The chapter closes with an overview of NGINX Plus.

Chapter 2, Basic NGINX Configuration, explores the essentials of NGINX configuration, from file syntax and structure to testing changes. Highlights include configuration syntax, base directives, and server testing strategies.

Chapter 3, Exploring the HTTP Configuration, focuses on the intricacies of the NGINX HTTP module, providing a comprehensive guide to directives and an in-depth exploration of the Location block. From the fundamentals of the HTTP Core module, through in-depth discussions of directives and variables, to mastery of the Location block, this chapter provides readers with the know-how to efficiently configure virtual hosts and understand the principles underlying NGINX's configuration structure.

Chapter 4, Exploring Module Configuration in NGINX, delivers an in-depth guide to NGINX's Rewrite module and a comprehensive overview of the open source edition's first-party modules. This chapter focuses on mastering "pretty URLs" with the Rewrite module, and exploring NGINX's vast selection of add-on modules.

Chapter 5, PHP and Python with NGINX, emphasizes configuring NGINX to work with dynamic PHP and Python scripts via FastCGI. Starting with a deep dive into the principles of FastCGI, it then walks through the process of configuring NGINX for efficient communication with PHP and Python applications, offering insights into using the FastCGI module for optimal server-side script integration.

Chapter 6, NGINX as a Reverse Proxy, discusses NGINX's reverse proxy capabilities and dives into the details of the NGINX proxy module. Readers will learn how to navigate the proxy module, appreciate NGINX's strengths in managing modern web applications, and set up a reverse proxy configuration efficiently.

Chapter 7, Introduction to Load Balancing and Optimization, focuses on deploying NGINX to manage high-traffic websites on multiple servers. This chapter looks at NGINX's advanced load-balancing capabilities and the latest I/O optimization features, including the use of thread pools. It offers a comprehensive guide to setting up a balanced server architecture and improving I/O performance, crucial for the scalability and efficiency of large-scale web applications.

Chapter 8, NGINX within a Cloud Infrastructure, highlights the integration of NGINX and Docker in cloud setups, from Docker fundamentals to deploying NGINX with Docker Compose. Learn to optimize containerized applications and caching with NGINX as a central gateway.

Chapter 9, Fully Deploy, Manage, and Auto-Update NGINX with Ansible, explores efficient NGINX deployment across multiple servers using Ansible. Covering mass deployment strategies, the basics of configuration management, and crafting Ansible roles for NGINX, this chapter will help with large-scale server management. It concludes with automating NGINX updates for effortless maintenance to deliver a robust and secure NGINX fleet.

Chapter 10, Case Studies, dives into the deployment of secure websites and applications with detailed case studies. Master SSL with `acme.sh`, adopt HTTP/2, configure WordPress and integrate NextCloud with Docker. Gain skills in deploying secure sites, configuring NGINX for various scenarios and using NGINX to enhance Docker environments.

Chapter 11, Troubleshooting, navigates through NGINX debugging with essential tips and tools for identifying issues. Covering general troubleshooting advice, log parsing, and common challenges such as install errors and 403 Forbidden pages, this chapter lays out a clear path to resolving configuration mistakes and understanding block priorities. Learn to adjust configurations for effective debugging and recognize frequent NGINX configuration errors.

To get the most out of this book

Operating system requirements	Software/hardware covered in the book
Linux (preferably Debian, Ubuntu, Fedora, or RHEL)	NGINX
	Ansible
	Docker

If you are using the digital version of this book, we advise you to type the code yourself or access the code from the book's GitHub repository (a link is available in the next section). Doing so will help you avoid any potential errors related to the copying and pasting of code.

Download the example code files

You can download the example code files for this book from GitHub at https://github.com/PacktPublishing/nginx-http-server_fifth-edition. If there's an update to the code, it will be updated in the GitHub repository.

We also have other code bundles from our rich catalog of books and videos available at <https://github.com/PacktPublishing/>. Check them out!

Conventions used

There are a number of text conventions used throughout this book.

Code in text: Indicates code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles. Here is an example: “Mount the downloaded `WebStorm-10*.dmg` disk image file as another disk in your system.”

A block of code is set as follows:

```
[nginx]
name=nginx repo
baseurl=http://nginx.org/packages/OS/OSRELEASE/$basearch/
pgpcheck=0
enabled=1
```

Any command-line input or output is written as follows:

```
apt update
apt install nginx
```

Bold: Indicates a new term, an important word, or words that you see onscreen. For instance, words in menus or dialog boxes appear in **bold**. Here is an example: “Select **System info** from the **Administration** panel.”

Tips or important notes

Appear like this.

Get in touch

Feedback from our readers is always welcome.

General feedback: If you have questions about any aspect of this book, email us at `customer@packtpub.com` and mention the book title in the subject of your message.

Errata: Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you have found a mistake in this book, we would be grateful if you would report this to us. Please visit `www.packtpub.com/support/errata` and fill in the form.

Piracy: If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at copyright@packt.com with a link to the material.

If you are interested in becoming an author: If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, please visit authors.packtpub.com.

Share Your Thoughts

Once you've read *NGINX HTTP Server*, we'd love to hear your thoughts! Please click [here](#) to go straight to the Amazon review page for this book and share your feedback.

Your review is important to us and the tech community and will help us make sure we're delivering excellent quality content.

Download a free PDF copy of this book

Thanks for purchasing this book!

Do you like to read on the go but are unable to carry your print books everywhere?

Is your eBook purchase not compatible with the device of your choice?

Don't worry, now with every Packt book you get a DRM-free PDF version of that book at no cost.

Read anywhere, any place, on any device. Search, copy, and paste code from your favorite technical books directly into your application.

The perks don't stop there, you can get exclusive access to discounts, newsletters, and great free content in your inbox daily

Follow these simple steps to get the benefits:

1. Scan the QR code or visit the link below



<https://packt.link/free-ebook/9781835469873>

2. Submit your proof of purchase
3. That's it! We'll send your free PDF and other benefits to your email directly

Part 1: Begin with NGINX

In this part, you will be introduced to the foundations of NGINX. This section provides the basis for understanding NGINX and its capabilities by exploring the installation and the baseline configuration of the web server. You will get practical information on the efficient configuration for NGINX, which will prepare you for the advanced configuration and optimization techniques covered in the next parts.

This part includes the following chapters:

- *Chapter 1, Downloading and Installing NGINX*
- *Chapter 2, Basic NGINX Configuration*

Downloading and Installing NGINX

NGINX (pronounced *engine-x*) has become the leader among web servers ever since it came out 20 years ago. Back in 2004, its main goal was to outperform Apache, and today, NGINX is outperforming every web server when it comes to high-traffic websites or security. Throughout this book, we will discover and learn how to use NGINX, step by step. We will cover many subjects to have a configuration tailored to everyone's needs.

In this first chapter, we will proceed with the necessary steps toward establishing a functional setup of NGINX. This moment is crucial for the smooth functioning of your web server—there are some required libraries and tools for installing the web server, some parameters that you will have to decide upon when compiling the binaries, and there may also be some configuration changes to perform on your system.

By the end of this chapter, you will have installed NGINX either through a public repository or by compiling a custom version embedding all the extra modules you might need.

This chapter covers the following:

- Installing NGINX via package managers
- Downloading and installing the prerequisites for compiling NGINX binaries
- Downloading a suitable version of the NGINX source code
- Configuring NGINX compile-time options
- Controlling the application with a `unit service` file
- Configuring the system to launch NGINX automatically on startup
- A quick overview of the possibilities offered by NGINX Plus

...

Installing NGINX via package managers

The quickest, and easiest, way to install NGINX is to simply use your OS-provided version. Most of the time, these are kept fairly updated; however, for some Linux distributions focusing on stability, you may only have older versions of NGINX available. Sometimes, your Linux distribution may provide multiple versions of NGINX with different compile flags.

In general, before embarking on a more complex journey, we should check whether we can use the easy solution. For Red Hat Linux-based operating systems, we need to enable the EPEL repo first and then do the same:

```
yum install epel-release
yum search nginx
yum info PACKAGE_NAME
yum install PACKAGE_NAME
```

For a Debian-based operating system, we first find the NGINX compiles available and then get the information for the one we want:

```
apt-cache search nginx
apt-cache show PACKAGE_NAME
apt install PACKAGE_NAME
```

If the version provided is current enough, then you're ready to configure NGINX in the next chapter.

If the version provided by your distribution is too old, then NGINX provides packages for **RHEL/CentOS distributions** as well as **Debian/Ubuntu distributions**. We encourage you to visit the official NGINX website to make sure the version given by your distribution isn't outdated.

NGINX-provided packages

To set up a yum repository for RHEL/CentOS, create a file named `/etc/yum.repos.d/nginx.repo` with the following contents:

```
[nginx]
name=nginx repo
baseurl=http://nginx.org/packages/OS/OSRELEASE/$basearch/
gpgcheck=0
enabled=1
```

Replace `OS` with `rhel` or `centos`, depending on the distribution used, and `OSRELEASE` with `8` or `9`, for versions `8.x` or `9.x`, respectively. Afterward, NGINX can now be installed with `yum`:

```
yum install nginx
```

For Debian-based distributions, we need to first use their signing key to authenticate the package signatures. Download the following file first from http://nginx.org/keys/nginx_signing.key.

Then, run the following command:

```
sudo apt-key add nginx_signing.key
```

With the key added, we can now add the NGINX repository to `sources.list` found in `/etc/apt/sources.list`. For Debian, we add the following lines:

```
deb http://nginx.org/packages/debian/ codename nginx
deb-src http://nginx.org/packages/debian/ codename nginx
```

Here, `codename` is either `trixie` or `bookworm`, depending on your version of Debian. For Ubuntu, we use the following dependencies:

```
deb http://nginx.org/packages/ubuntu/ codename nginx
deb-src http://nginx.org/packages/ubuntu/ codename nginx
```

Here, `codename` is either `noble`, `focal`, or `bionic`, depending on your version of Ubuntu. Finally, we can install NGINX with the `apt` command:

```
apt update
apt install nginx
```

Now that we have learned how to install NGINX from repositories, let's have a look at how we can compile it from the source and benefit from having custom modules that are not provided with the default NGINX.

Compiling NGINX from source

There are situations where compiling NGINX from source is preferable. It gives us the most flexibility regarding modules, so we can customize better for our intended usage. For example, we could compile a very lean version for embedded hardware.

Additionally, we can make sure we use the latest version of NGINX and have all new features available to us. Keep in mind, though, that when installing software from source, you are responsible for keeping it updated. NGINX, just like every other piece of software, sometimes finds security issues that it needs to address. An OS package is much easier to update than a source installation but, so long as you're aware of the need to maintain it yourself, there is absolutely no problem.

Depending on the optional modules that you select at compile time, you will perhaps need different prerequisites. We will guide you through the process of installing the most common ones, such as **GCC**, **PCRE**, **zlib**, and **OpenSSL**.

Installing GNU Compiler Collection

NGINX is a program written in C, so you will first need to install a compiler tool such as the **GNU Compiler Collection (GCC)** on your system. GCC may already be present on your system, but if that is not the case, you will have to install it before going any further.

Note

GCC is a collection of free open source compilers for various languages – C, C++, Java, Ada, Fortran, and so on. It is the most commonly used compiler suite in the Linux world, and Windows versions are also available. A vast number of processors are supported, such as x86, AMD64, PowerPC, ARM, MIPS, and more.

Follow these steps to install GCC:

1. First, make sure it isn't already installed on your system:

```
[user@server ~]$ gcc
```

If you get the following output, it means that GCC is correctly installed on your system and you can skip to the next section:

```
gcc: no input files
```

If you receive the following message, you will have to proceed with the installation of the compiler:

```
~bash: gcc: command not found
```

2. GCC can be installed using the default repositories of your package manager. Depending on your distribution, the package manager will be vary-yum for a Red Hat Linux-based distribution, apt for Debian and Ubuntu, yast for SUSE Linux, and so on. Here is the typical way to proceed with the download and installation of the GCC package:

```
[root@server ~]# yum groupinstall "Development Tools"
```

3. If you use apt, execute the following command:

```
[root@server ~]# apt-get install build-essentials
```

If you use another package manager with a different syntax, you will probably find the documentation with the man utility. Either way, your package manager should be able to download and install GCC correctly, after having resolved dependencies automatically.

Note that the apt and yum commands will not only install GCC; they will also proceed to download and install all common requirements for building applications from source, such as code headers and other compilation tools.

The PCRE library

The **Perl Compatible Regular Expressions (PCRE)** library is required for compiling NGINX. The rewrite and HTTP core modules of NGINX use PCRE for the syntax of their regular expressions, as we will discover in later chapters. You will need to install two packages—`pcre` and `pcre-devel`. The first one provides the compiled version of the library, whereas the second one provides development headers and sources for compiling projects, which are required in our case.

Here are some example commands that you can run in order to install both packages.

Using `yum`, execute the following command:

```
[root@server ~]# yum install pcre pcre-devel
```

Or you can install all PCRE-related packages using the following command:

```
[root@server ~]# yum install pcre*
```

If you use `apt`, use the following command:

```
[root@server ~]# apt install libpcre3 libpcre3-dev
```

If these packages are already installed on your system, you will receive a message saying something like `nothing to do`; in other words, the package manager did not install or update any component:

```
root@server:~# apt install libpcre3 libpcre3-dev
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
libpcre3 is already the newest version (2:8.39-13ubuntu0.22.04.1).
libpcre3-dev is already the newest version (2:8.39-13ubuntu0.22.04.1).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@server:~# _
```

Figure 1.1: APT showing the PCRE library is already installed.

The preceding output signifies that both the components, `libpcre3` and `libpcre3-dev`, are already present in the system.

The zlib library

The `zlib` library provides developers with compression algorithms. It is required for the use of `.gzip` compression in various modules of NGINX. Again, you can use your package manager to install this component as it is part of the default repositories. Similar to PCRE, you will need the `zlib` library and its associated `zlib-dev` component as well.