

[Open in app ↗](#)[Sign up](#)[Sign in](#)**Medium**

Search



How to Set Up a File Download Server on Ubuntu with Apache

Jobin J · [Follow](#)

3 min read · May 10, 2024

[Listen](#)[Share](#)

Introduction

In today's digital age, the need for easy and secure file sharing is more important than ever. Whether you're sharing documents, media files, or software packages, having a dedicated file download server can streamline the process and improve efficiency. In this guide, we'll walk you through the step-by-step process of setting up a file download server on Ubuntu using the Apache web server.

Prerequisites

Before we begin, make sure you have the following:

- Access to an Ubuntu server instance (this can be a physical server or a virtual machine).
- Basic familiarity with the Linux command line.
- Administrative privileges to install software and configure system settings.

Step 1: Install Apache

The first step is to install the Apache web server. Open a terminal and run the following command:

```
sudo apt update  
sudo apt install apache2
```

This will install Apache on your Ubuntu server.

Step 2: Configure Firewall

Next, we need to configure the firewall to allow incoming traffic to the Apache web server. If you're using UFW (Uncomplicated Firewall), you can allow Apache traffic by running the following command:

```
sudo ufw allow 'Apache'
```

This will allow HTTP and HTTPS traffic to reach your Apache server.

Step 3: Create a Directory for Files

Now, let's create a directory where our files will be stored. You can choose any directory you like, but for this example, we'll use the default Apache web root directory:

```
sudo mkdir /var/www/html/downloads
```

This command creates a new directory named "downloads" inside the Apache web root directory.

Step 4: Upload Files

Upload the files you want to share to the “downloads” directory using your preferred method. You can use SCP, SFTP, or any other file transfer method you’re comfortable with.

For demonstration purposes, let’s create some sample files in the “downloads” directory. You can create sample files using the following command:

```
sudo touch /var/www/html/downloads/demo.txt
```

This command creates a new file named “demo.txt” inside the “downloads” directory. You can repeat this step to create multiple sample files if needed.

Step 5: Configure Apache

Next, we need to configure Apache to serve files from the “downloads” directory. Open the Apache configuration file in a text editor:

```
sudo nano /etc/apache2/sites-available/000-default.conf
```

Inside the `<virtualHost>` block, add the following lines:

```
<Directory /var/www/html/downloads>
    Options +Indexes
    AllowOverride None
    Require all granted
</Directory>
```

Save the file and exit the text editor.

Step 6: Restart Apache

To apply the changes we’ve made to the Apache configuration, restart the Apache web server:

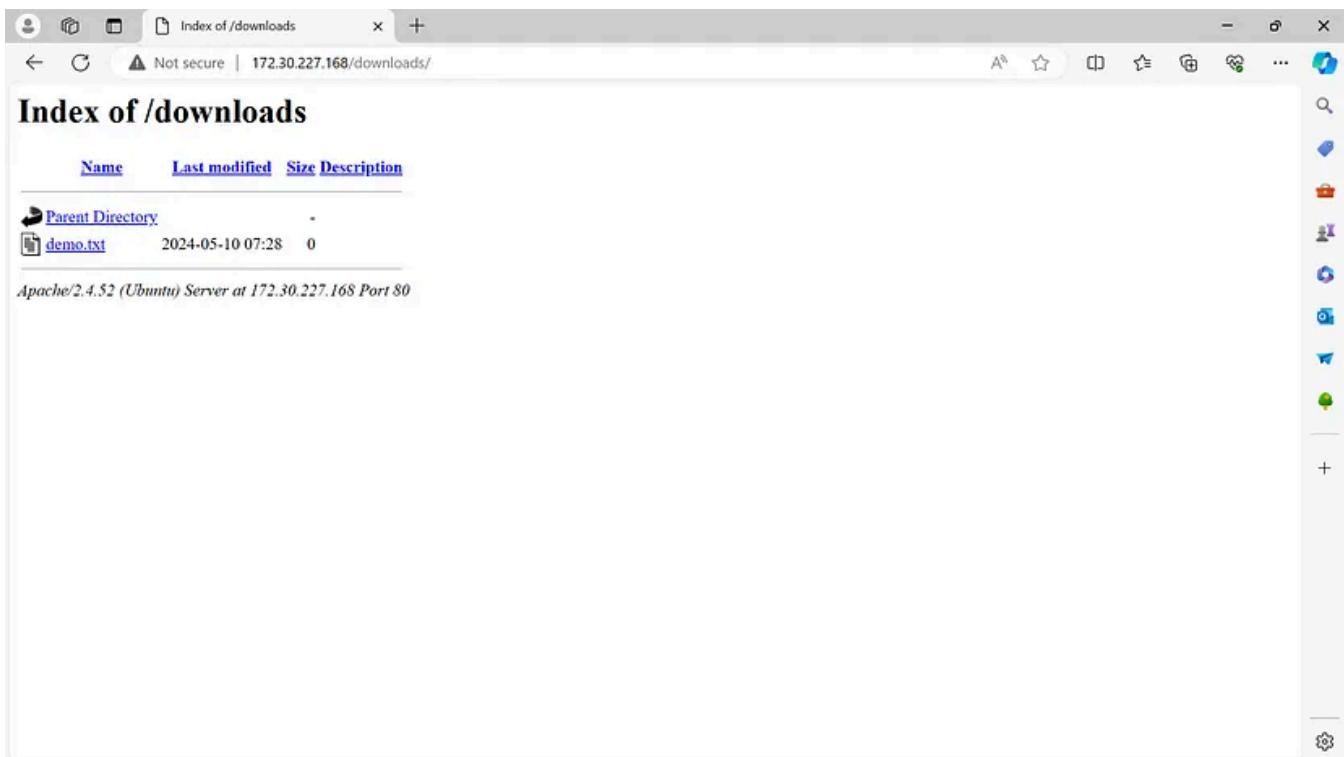
```
sudo systemctl restart apache2
```

Step 7: Access Files

You can now access the files hosted on your Ubuntu server by navigating to the following URL in your web browser:

```
http://your_server_ip/downloads
```

Replace “your_server_ip” with the IP address of your Ubuntu server. You should see a list of files available for download.



Conclusion

Setting up a file download server on Ubuntu with Apache is a straightforward process that can greatly simplify the file sharing process. By following the steps outlined in this guide, you can create an efficient platform for sharing files with colleagues, clients, or friends.

(NOTE : While the setup described provides a basic level of file sharing functionality, it may not be considered fully secure for all use cases.)

Happy file sharing!

[File Server](#)[Ftp](#)[Ssh](#)[Ubuntu](#)[Apache](#)[Follow](#)

Written by Jobin J

7 Followers

Software Developer

More from Jobin J



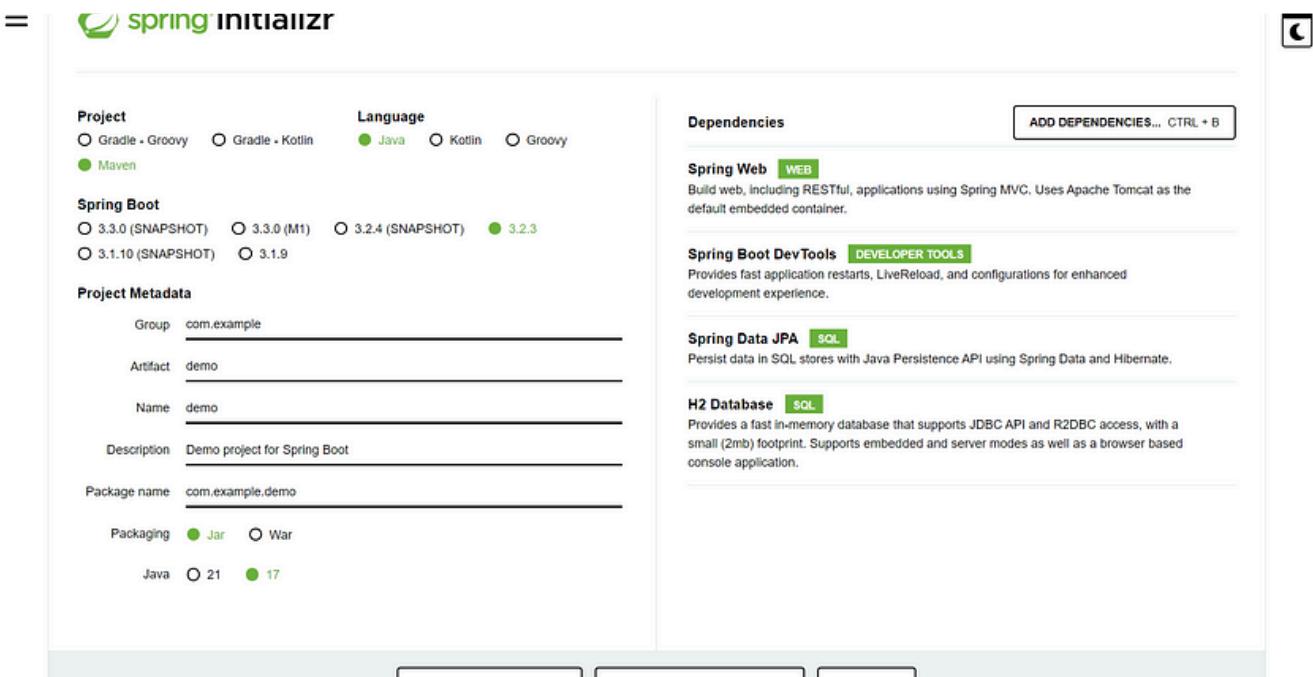
How to Run Ubuntu as a Docker Container

Docker is a popular platform for developing, shipping, and running applications in containers. Containers allow you to package an...

Apr 3  2 Jobin J

Accessing Google Sheet Data with Python: A Practical Guide using the Google Sheets API

Create a Google Cloud Platform (GCP) project

Apr 27  2


Project

- Gradle - Groovy
- Gradle - Kotlin
- Java
- Kotlin
- Groovy

Language

- Maven

Spring Boot

- 3.3.0 (SNAPSHOT)
- 3.3.0 (M1)
- 3.2.4 (SNAPSHOT)
- 3.2.3

Project Metadata

Group	com.example
Artifact	demo
Name	demo
Description	Demo project for Spring Boot
Package name	com.example.demo
Packaging	<input checked="" type="radio"/> Jar
Java	<input type="radio"/> 21 <input checked="" type="radio"/> 17

Dependencies

- Spring Web** WEB
- Spring Boot DevTools** DEVELOPER TOOLS
- Spring Data JPA** SQL
- H2 Database** SQL

ADD DEPENDENCIES... CTRL + B



Jobin J

Getting Started | Building an Application with Spring Boot | Eclipse IDE

Step 1: Create Project

Mar 2



Jobin J

Adding External Python Libraries to AWS Lambda: A Step-by-Step Guide

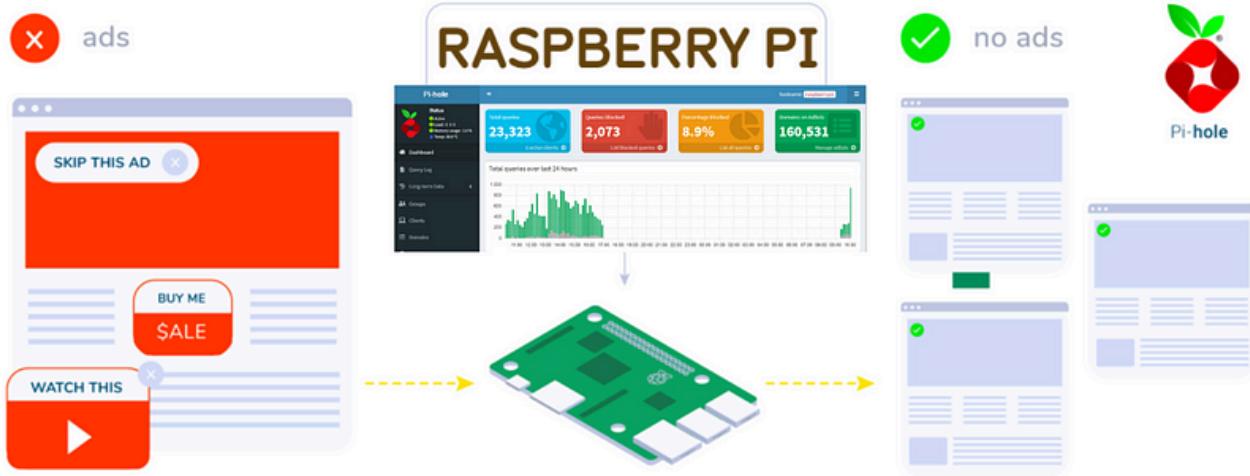
AWS Lambda allows you to run code without provisioning or managing servers, making it an ideal choice for serverless applications. However...

Apr 27



See all from Jobin J

Recommended from Medium

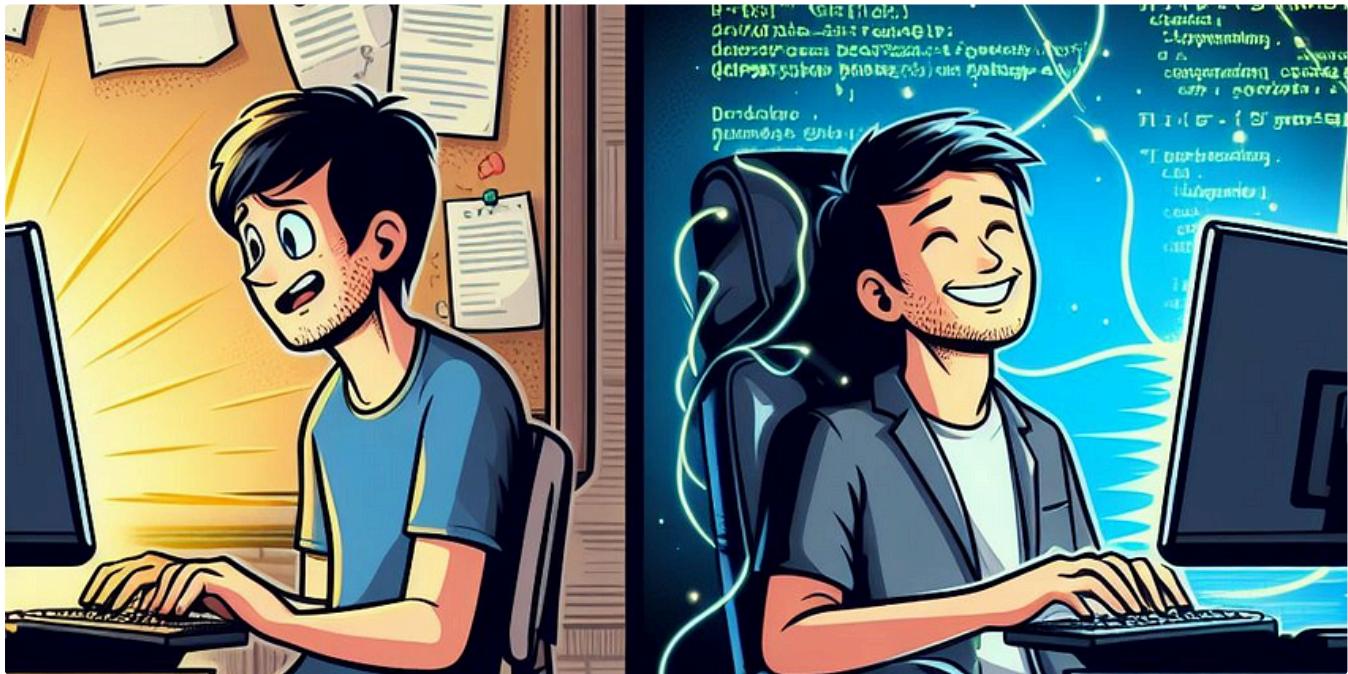


 Richard Taujenis in Stackademic

Transform Your Raspberry Pi into a Network-wide Ad Blocker

Transform your Raspberry Pi into a powerful network-wide ad blocker by installing Pi-Hole, allowing you to filter out ads and trackers.

★ Aug 3 ⌂ 1



 Abhay Parashar in The Pythoneers

17 Mindblowing Python Automation Scripts I Use Everyday

Scripts That Increased My Productivity and Performance

1d ago 6.9K 59



Lists



Staff Picks

718 stories · 1251 saves



Stories to Help You Level-Up at Work

19 stories · 763 saves



Self-Improvement 101

20 stories · 2622 saves



Productivity 101

20 stories · 2255 saves



 Manyi

Generate a SSH key for connecting to a server

Here is a step-by-step guide to generate a key for SSH connection.

1d ago Jul 19



Wildcard SSL Certificate

Nginx configuration



Eurico Paes

Configure Nginx with a Wildcard SSL Certificate (Let's Encrypt)

An SSL certificate for www.domain.com does not automatically work for blog.domain.com unless it is explicitly designed to do so. There are...

Jun 22 1



amazon.com

Software Development Engineer

Seattle, WA

Mar. 2020 – May 2021

- Developed Amazon checkout and payment services to handle traffic of 10 Million daily global transactions
- Integrated Iframes for credit cards and bank accounts to secure 80% of all consumer traffic and prevent CSRF, cross-site scripting, and cookie-jacking
- Led Your Transactions implementation for JavaScript front-end framework to showcase consumer transactions and reduce call center costs by \$25 Million
- Recovered Saudi Arabia checkout failure impacting 4000+ customers due to incorrect GET form redirection

Projects

NinjaPrep.io (React)

- Platform to offer coding problem practice with built in code editor and written + video solutions in React
- Utilized Nginx to reverse proxy IP address on Digital Ocean hosts
- Developed using Styled-Components for 95% CSS styling to ensure proper CSS scoping
- Implemented Docker with Seccomp to safely run user submitted code with < 2.2s runtime

HeatMap (JavaScript)

- Visualized Google Takeout location data of location history using Google Maps API and Google Maps heatmap code with React
- Included local file system storage to reliably handle 5mb of location history data
- Implemented Express to include routing between pages and jQuery to parse Google Map and implement heatmap overlay



Alexander Nguyen in Level Up Coding

The resume that got a software engineer a \$300,000 job at Google.

1-page. Well-formatted.

Jun 1

18.9K

314



Wensen Ma

Step-by-Step Guide to Setting Up an OpenVPN Server on Ubuntu

Creating your own Virtual Private Network (VPN) can greatly enhance your internet security by encrypting your internet traffic and hiding...

Mar 23

2

[See more recommendations](#)