



Get unlimited access to all of Medium. [Become a member](#)

Automating Laravel CI/CD with GitHub Actions on Namecheap



Daniel Mabadeje · [Follow](#)

6 min read · 13 hours ago



Listen



Share



More



GitHub Actions

Who is this article for?

Have you had some struggles hosting your Laravel application on namecheap? Or you have been looking for a way to automate your deployments to your namecheap server? This article is for you.

Lets talk about Github actions a bit 😊

GitHub Actions is a powerful tool for automating various workflows and tasks within the GitHub ecosystem. Here are some of the key benefits of using GitHub Actions:

1. **Continuous Integration and Deployment (CI/CD):** GitHub Actions enables you to set up automated CI/CD pipelines, allowing you to build, test, and deploy your

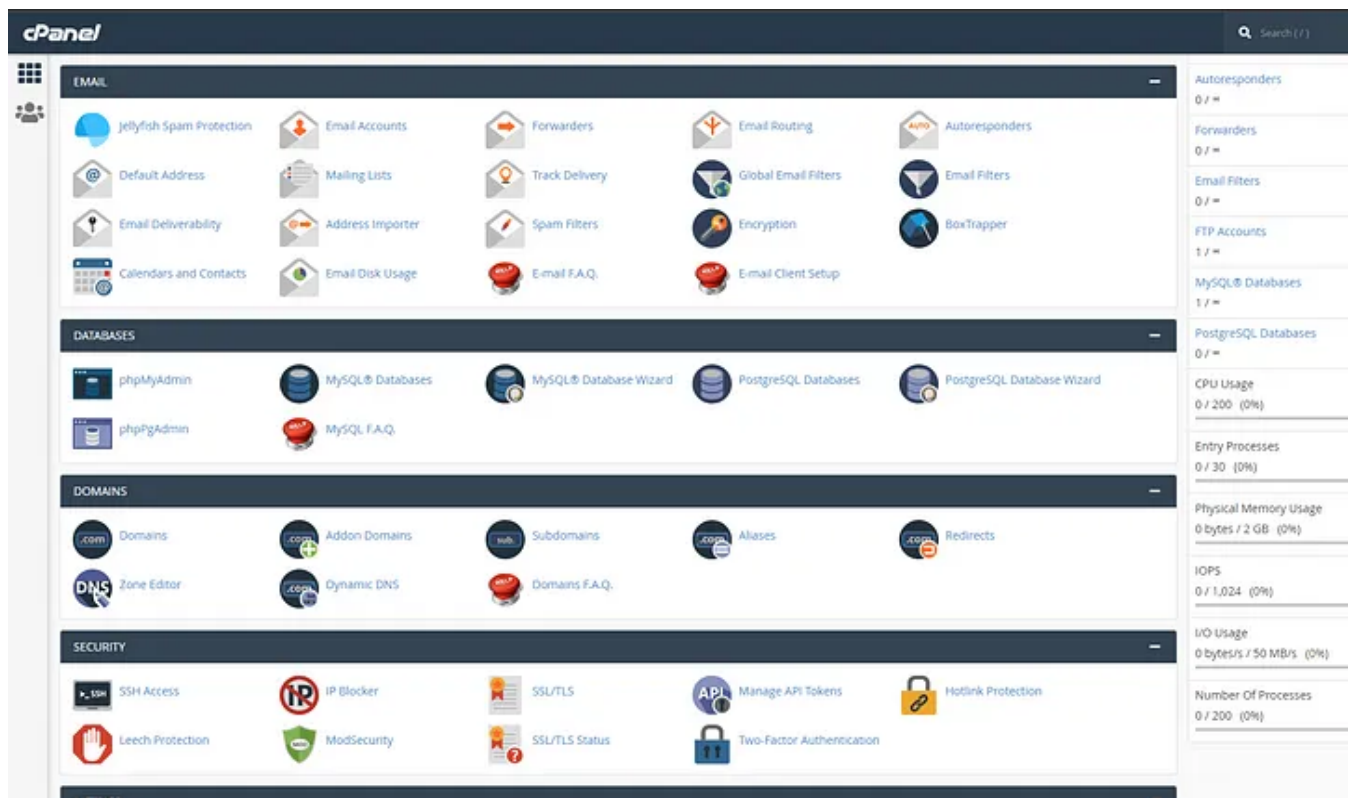
applications with ease. You can define custom workflows that trigger actions based on events like pushes, pull requests, or scheduled intervals. (This is what we will be focusing in this article)

2. **Flexibility and Customization:** GitHub Actions provides a flexible and customizable platform where you can define workflows using YAML syntax. You have granular control over the steps, dependencies, and conditions of your actions, allowing you to create tailored workflows that fit your specific needs.
3. **Seamless Integration with GitHub:** Since GitHub Actions is tightly integrated with the GitHub platform, it offers seamless integration with your repositories, issues, pull requests, and other GitHub features. You can leverage this integration to create workflows that interact with your repository, update issues, trigger notifications, and more.
4. **Scalability:** GitHub Actions is designed to scale with your needs. You can run workflows on different types of virtual machines and specify resource allocation for each job. Additionally, parallel and matrix strategies allow you to distribute workloads across multiple machines, enabling faster execution times.
5. **Cross-Platform Support:** GitHub Actions supports a wide range of platforms and operating systems, including Linux, macOS, and Windows. You can test your applications across different environments, ensuring compatibility and reliability.
6. **Cost-Efficient:** GitHub Actions provides a generous amount of free usage every month, allowing you to automate your workflows without incurring additional costs. If you require more resources or have high-demand workflows, you can choose from various pricing options that align with your needs.
7. **Vast Ecosystem and Community Support:** GitHub Actions has a large ecosystem of pre-built actions that you can use to perform common tasks. These actions, created by both GitHub and the community, cover a wide range of functionalities, from building and testing applications to deploying to various platforms. You can also create and share your own actions with others.

With all these mentioned, It is safe to say that Github actions is the real IDAN (magic but colloquially means boss.) 😄

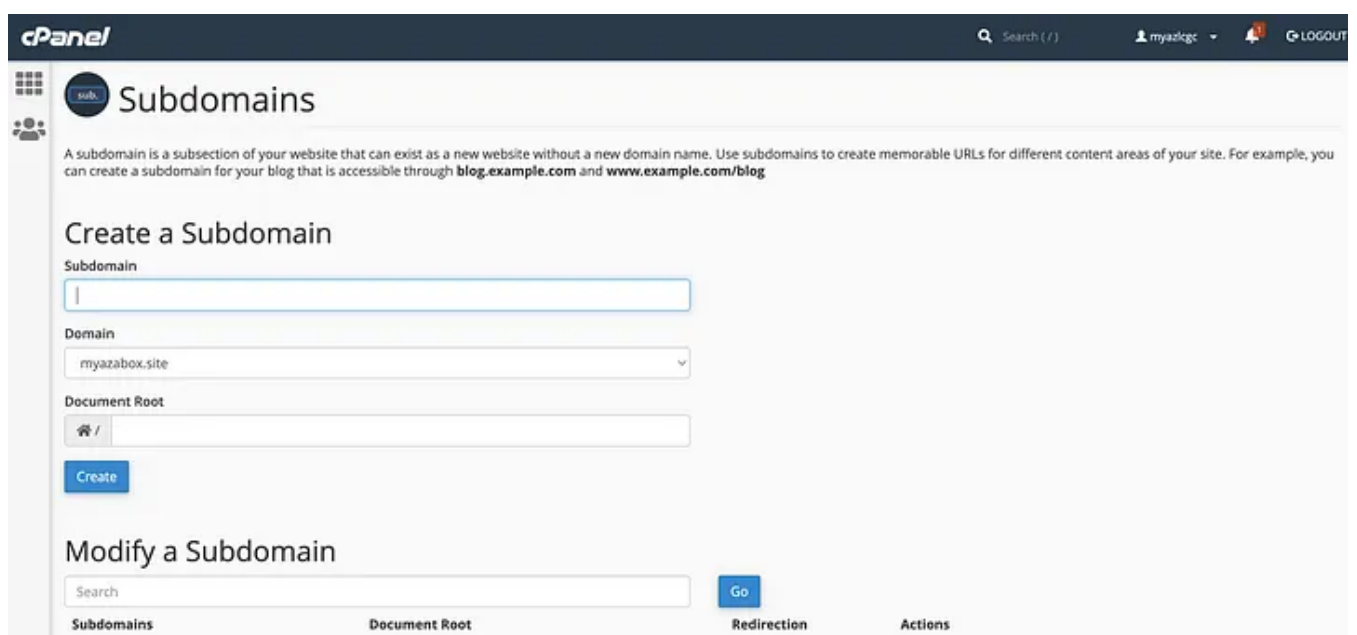
So How Do I Setup this Github Actions?

So first of all lets head to our cpanel, you shouls see something like this.



This particular step here should be followed if you want to deploy to a sub domain. If you want to deploy to the main domain itself, you may skip this step

Click on Subdomains



Fill in the details for your sub domain and click on create.

This would automatically create the subdomain and also create the document root.

Sidenote: I prefer to use the subdomain I created as the actual document root. So I can have my subdomain field as danielsub, my document root would be danielsub.myazabox.site.

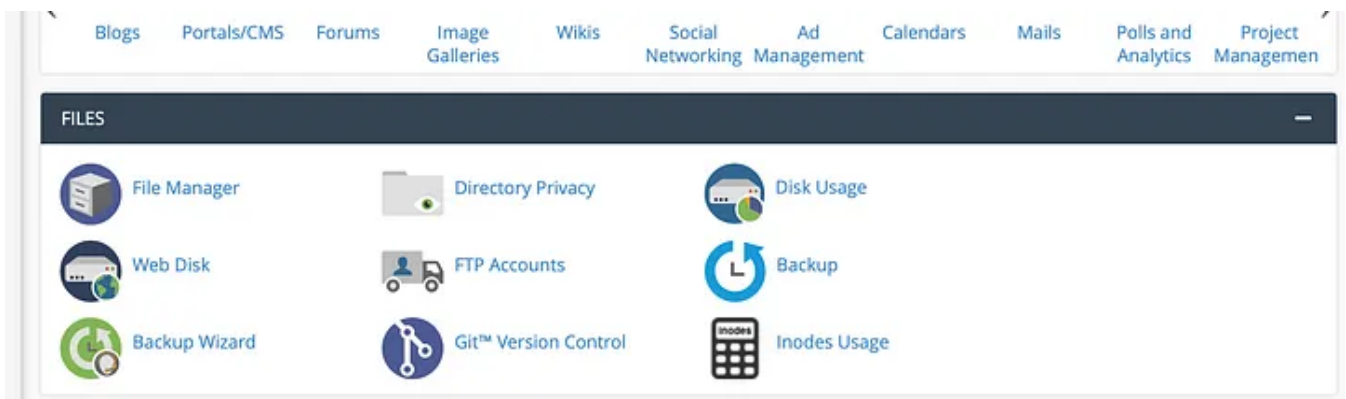
After we have successfully created the sub domain, we create an **FTP** account

What is an FTP Account? 🤔

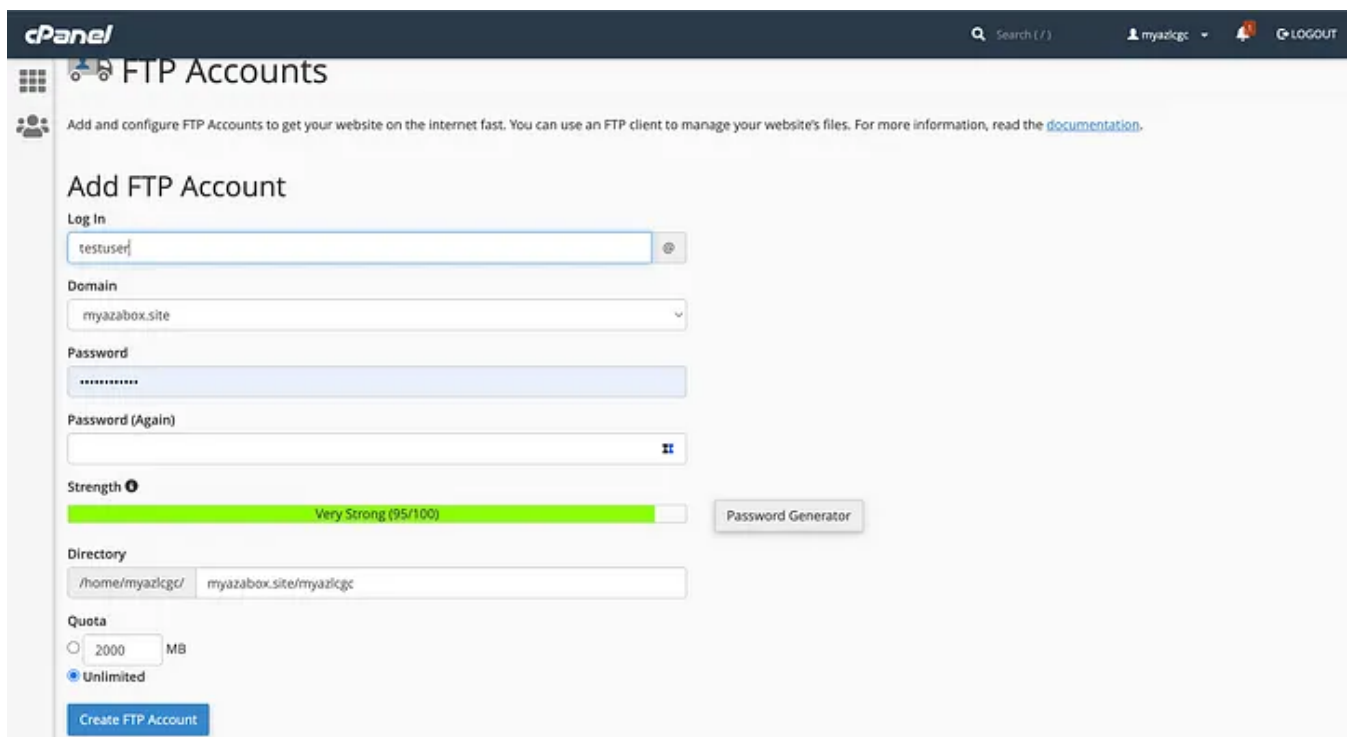
According to Techopedia, A file transfer protocol account (FTP account) is a type of user account that enables the transfer of files with a host computer by using FTP services.

So basically we are creating an account to facilitate file transfer to a server.

On your CPanel, Navigate to the files section as shown in the image



Click on FTP Accounts. At this point you should have something like this.



cPanel Search (/) myazlhc LOGOUT

FTP Accounts

Add and configure FTP Accounts to get your website on the internet fast. You can use an FTP client to manage your website's files. For more information, read the [documentation](#).

Add FTP Account

Log In:

Domain:

Password:

Password (Again):

Strength: Very Strong (95/100) [Password Generator](#)

Directory:

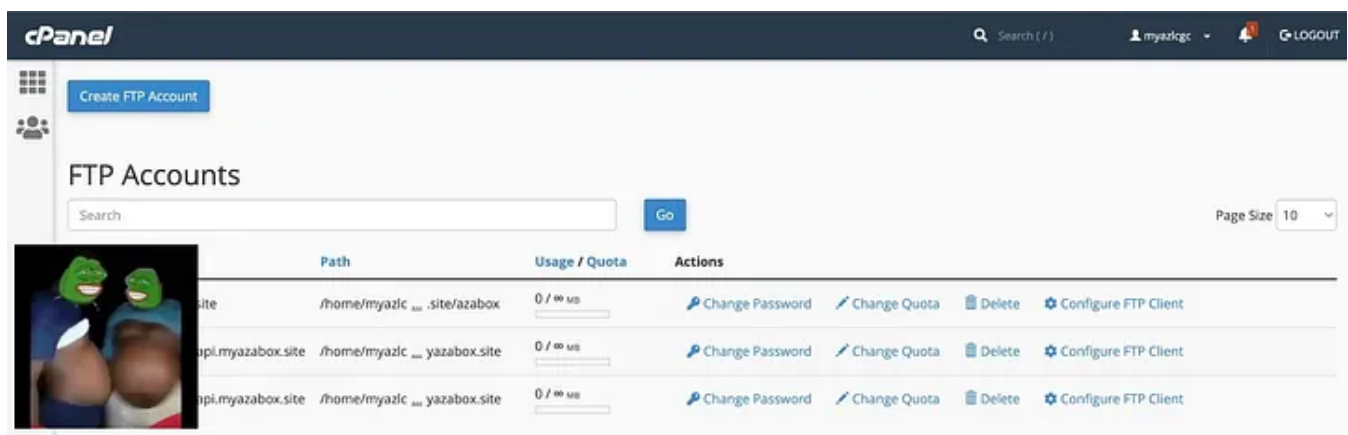
Quota: ☐ 2000 MB ☒ Unlimited

[Create FTP Account](#)

Put in your desired login details and ensure you stored those details properly in maybe your notes or somewhere safe (you would need them). Also select domain you want to assign the ftp account to. In this case it would be the subdomain that we created recently.

When you are done, you can click on the “Create FTP Account” button.

If you look below, you should see a new ftp account created as shown below



cPanel Search (/) myazlhc LOGOUT

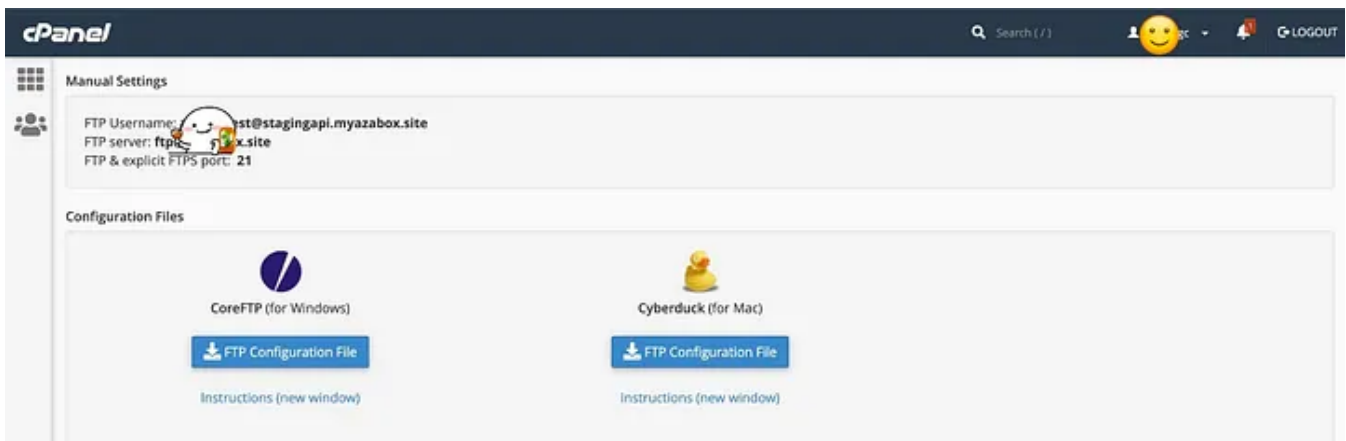
[Create FTP Account](#)

FTP Accounts

Search [Go](#) Page Size 10

	Path	Usage / Quota	Actions
site	/home/myazlhc/site/azabox	0 / ∞ MB	Change Password Change Quota Delete Configure FTP Client
api.myazabox.site	/home/myazlhc/api.yazabox.site	0 / ∞ MB	Change Password Change Quota Delete Configure FTP Client
api.myazabox.site	/home/myazlhc/api.yazabox.site	0 / ∞ MB	Change Password Change Quota Delete Configure FTP Client

to view your ftp details like Username, server details and in some cases Port. click on “Configure FTP Client”

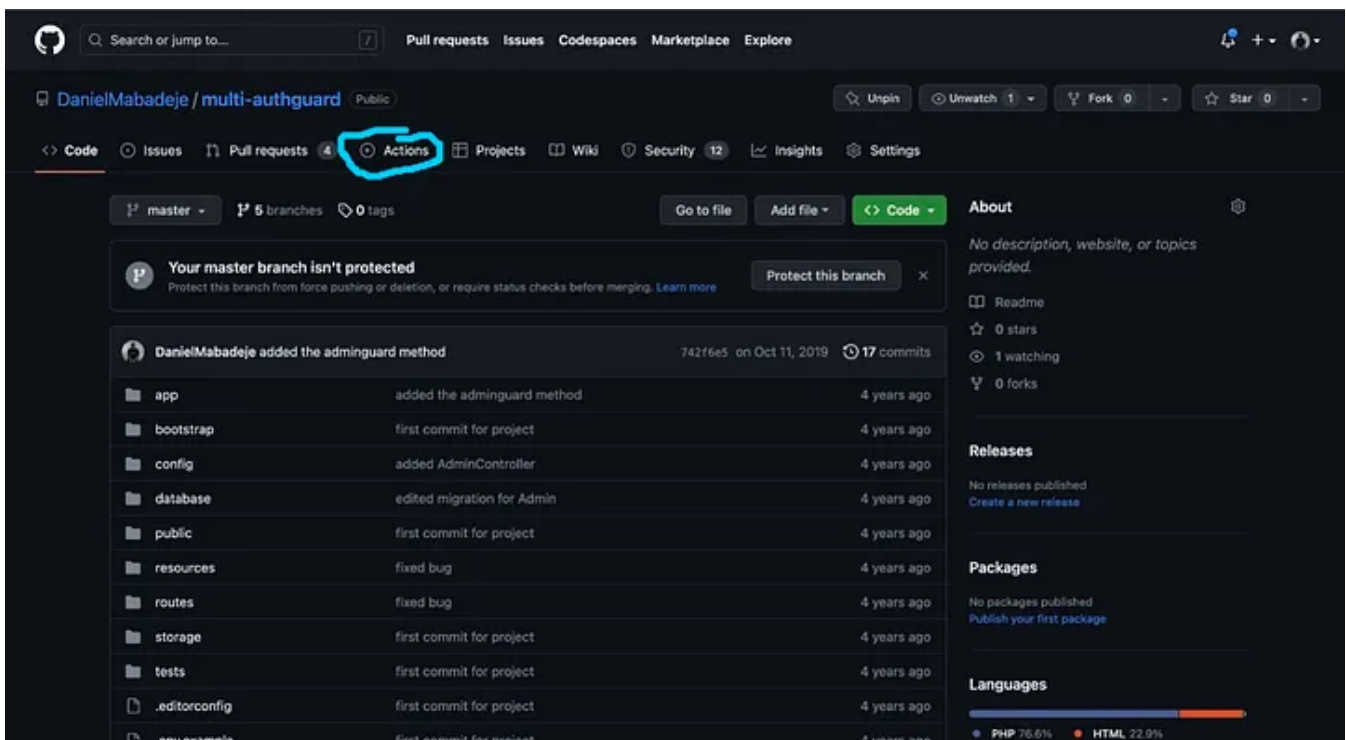


Now we are done with CPANEL, lets move over to github and perform this magic.

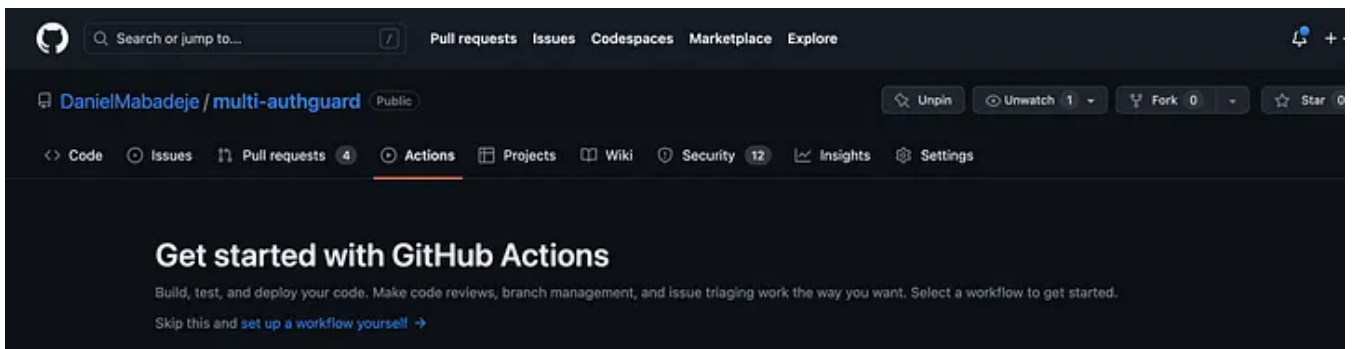
At this point I believe you already have your Laravel project pushed to [Github](#). If this has not been done, ensure you have pushed your laravel project to your repository. (This deployment procedure works for both personal and organisation account).

So for the purpose of this article, I will use a laravel application that I have pushed long ago.

Navigate to the repository that houses your laravel project, and click on the actions tab.



Click on “setup new workflow for yourself ->”



This would create a .yaml file for you to work with (mostly a main.yaml file)

Copy the code below and paste it in your main.yaml

```
name: Deploy Laravel Project on push
on:
  push:
    branches:
      - main
jobs:
  web-deploy:
    name: Deploy
    runs-on: ubuntu-latest
    steps:
      - name: Get the latest code
        uses: actions/checkout@v2.3.2
      - uses: actions/setup-node@master
      - name: Copy .env
        run: php -r "file_exists('.env') || copy('.env.example', '.env');"
      - name: Install Dependencies
        run: composer update --ignore-platform-reqs
      - name: Generate key
        run: php artisan key:generate
      - name: Generate storage link
        run: php artisan storage:link
      - name: Directory Permissions
        run: chmod -R 777 storage bootstrap/cache
      - name: Sync files
        uses: SamKirkland/FTP-Deploy-Action@v4.3.4
        with:
          server: ftp.myazabox.site
          username: YOURUSERNAME@test.myazabox.site
          password: PASSWORD
```

You can use the code above in your main.yaml or you could go to [Samkirkland](#) to view other use cases to deploy your application with Github actions.

The code above basically checks if there is a new push to the branch you specified. (In this case the main branch). Runs a composer update, generates a key with the laravel commands and also links the storage.

After this is done, it uploads the files to the document directory in your CPanel. Do not forget to change the username, ftp and password field to their respective values which you saved earlier on. To take it a notch higher, you can store them as SECRETS on github.

It is Important to note that your first deployment may take a long time to sync its files as it copies the newly created vendor folder (which houses a lot of files).

One more thing, If you are using an Apache server, there may be need to create a .htaccess in your root directory. After the .htaccess file creation, add the following code.

```
# RewriteEngine on

# # serve existing files in the /public folder as if they were in /
# RewriteCond %{DOCUMENT_ROOT}public%{REQUEST_URI} -f
# RewriteRule (.+) /public/$1 [L]

# # route everything else to /public/index.php
# RewriteRule ^ /public/index.php [L]

# DirectoryIndex index.php
<IfModule mod_rewrite.c>

    RewriteEngine On
    # RewriteBase /public_html/

    RewriteRule ^$ public/index.php [L]

    RewriteRule ^((?!public/).*)$ public/$1 [L,NC]

</IfModule>
```

And Voila!!! 🎉 🎉 🎉 🎉 🎉

You have successfully deployed a laravel application to your cpanel through github actions.

Follow me on twitter : <https://twitter.com/mabadejedaniel>

[Follow](#)

Written by Daniel Mabadeje

8 Followers

I am a full stack developer passionate about solving problems. I am a Co host at PHPTalks as well.

More from Daniel Mabadeje



Daniel Mabadeje

Writing A Simple MVC PHP Framework

Introduction

16 min read · Mar 10, 2020

```
8 $function$
9
10 postgres=# select foo();
11 ERROR:  division by zero
12 CONTEXT:  SQL statement "SELECT 10/0"
13 PL/pgSQL function "foo" line 2 at RETURN
14 postgres=#
15
16 **** CREATE OR REPLACE FUNCTION public.foo()
17 **** RETURNS integer
18 **** LANGUAGE plpgsql
1 19 1 AS $function$ begin
2 20 2 return 10/0;
3 21 3 end;
22 **** $function$
```

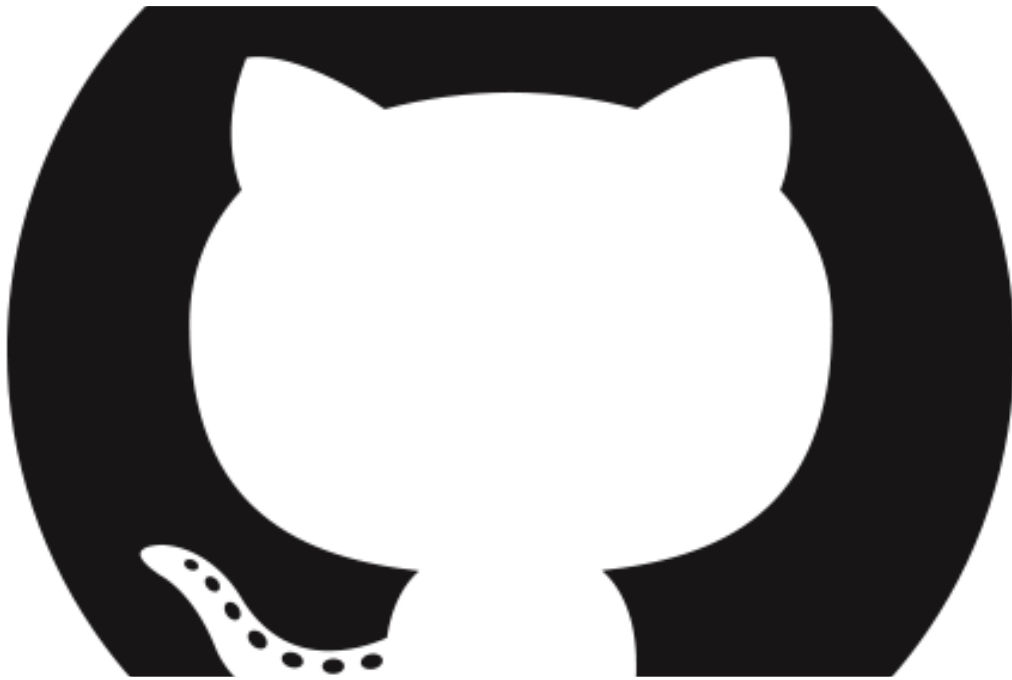


Daniel Mabadeje

How to create tables on Heroku Postgresql

So you want to deploy this your idea or project for demo on heroku. Probably, you are deploying from Github, then it's time for you to...

3 min read · Mar 3, 2021



Daniel Mabadeje

Git Commands Cheat sheet

So in this particular article i am going to be talking about some commands commonly used in git. But first of all, I will work you through...

4 min read · Jan 9, 2021



Daniel Mabadeje

How Web Browsers and Servers Communicate


Have you ever wondered how you are able to go to website on the browser and it views it or wondered why the website you opened in country...

3 min read · Oct 31, 2020



See all from Daniel Mabadeje


Recommended from Medium



Documentation

Laravel has wonderful documentation covering every aspect of the framework. Whether you are a newcomer or have prior experience with Laravel, we recommend reading our documentation from beginning to end.


[→](#)



Laracasts

Laracasts offers thousands of video tutorials on Laravel, PHP, and JavaScript development. Check them out, see for yourself, and massively level up your development skills in the process.


[→](#)



Laravel News

Laravel News is a community driven portal and newsletter aggregating all of the latest and most important news in the Laravel ecosystem, including new package releases and tutorials.

[→](#)



Vibrant Ecosystem

Laravel's robust library of first-party tools and libraries, such as [Forge](#), [Vapor](#), [Nova](#), and [Envoyer](#) help you take your projects to the next level. Pair them with powerful open source libraries like [Cashier](#), [Dusk](#), [Echo](#), [Horizon](#), [Sanctum](#), [Telescope](#), and more.



Balaji Dharma in Dev Genius

How to Upgrade From Laravel 9.x to Laravel 10.x

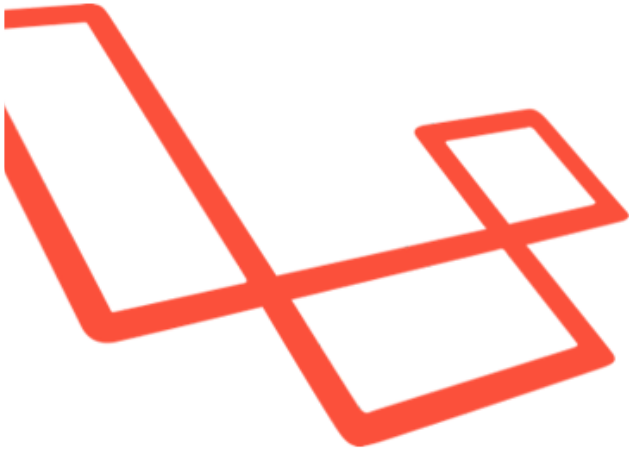
Laravel creates an admin panel from scratch—Laravel 10 Upgrade Guide— Part 23

★ · 5 min read · Feb 16



13





laravel



docker

 Dino Cajic  in Geek Culture

Setting Up Laravel 100% with Docker on a New Device

I frequently speak with individuals on how to setup work environments. There's still a large community that rejects Docker; I'm assuming...

★ · 4 min read · Dec 19, 2022



11



2



Lists



Leadership

30 stories · 28 saves



Blue-Green Deployment (CI/CD) Pipelines with Docker, GitHub...



Dmit in DevOps.dev

Blue-Green Deployment (CI/CD) Pipelines with Docker, GitHub, Jenkins and SonarQube

Blue-Green Deployment is a software deployment and delivery concept used to minimize downtime and risk during releases. The method works by...

★ · 5 min read · Feb 11



80



3



Laravel 10: New Features for Web Artisans



Dinesh Kumar

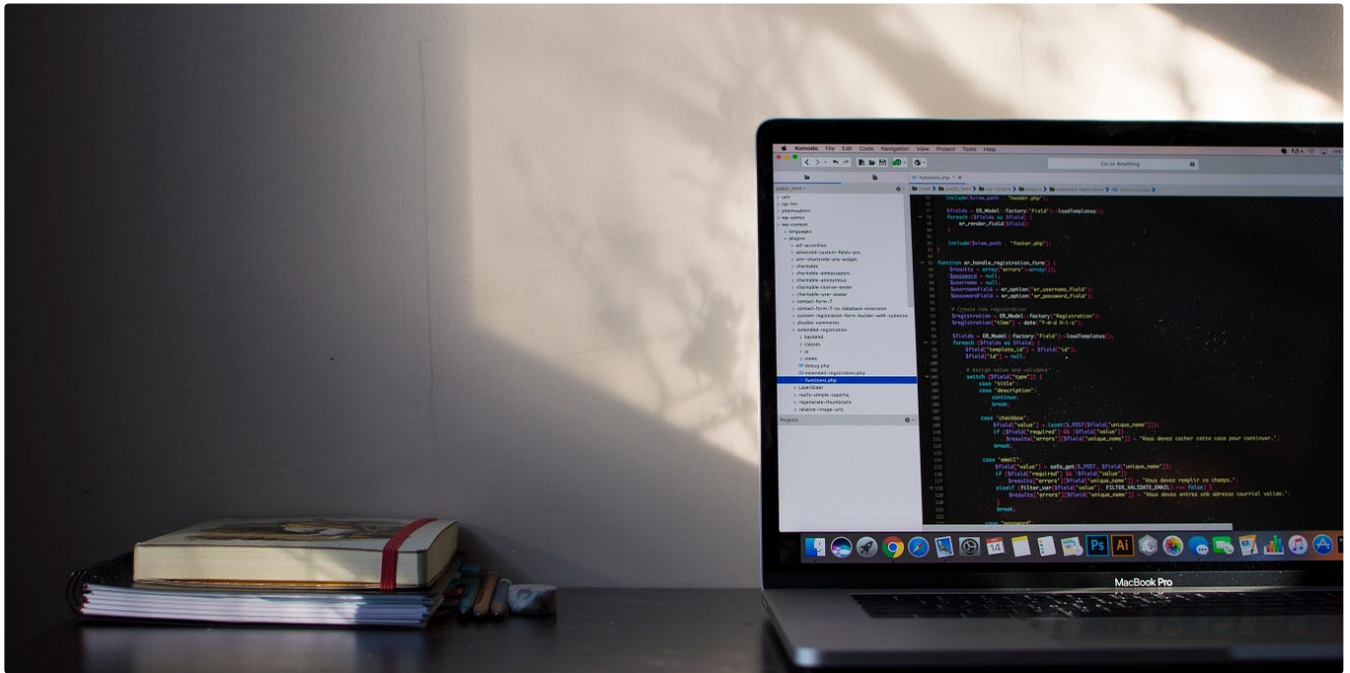
Exploring Laravel's Pennant Feature: A New Era of Interactive Development

Introduction

3 min read · Apr 13

👏 1 💬 2

🔖+ ⋮



 Muhammad Naufal Pratama

PHPMD: A PHP Code Smells Detector

Detect the messines in your PHP code without pulling you toward the messines itself

3 min read · May 8

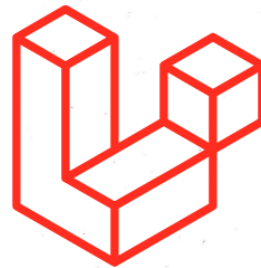
👏 3 💬

🔖+ ⋮



MUST-HAVE

LARAVEL PACKAGES FOR BUILDING POWERFUL WEB APPLICATIONS



Aswin

5 Must Have Laravel Packages for Building Powerful Web Applications

Laravel is a popular PHP framework known for its elegant syntax, powerful features, and robust ecosystem of packages. Whether you're...

4 min read · May 5



6



See more recommendations