

INSTALL

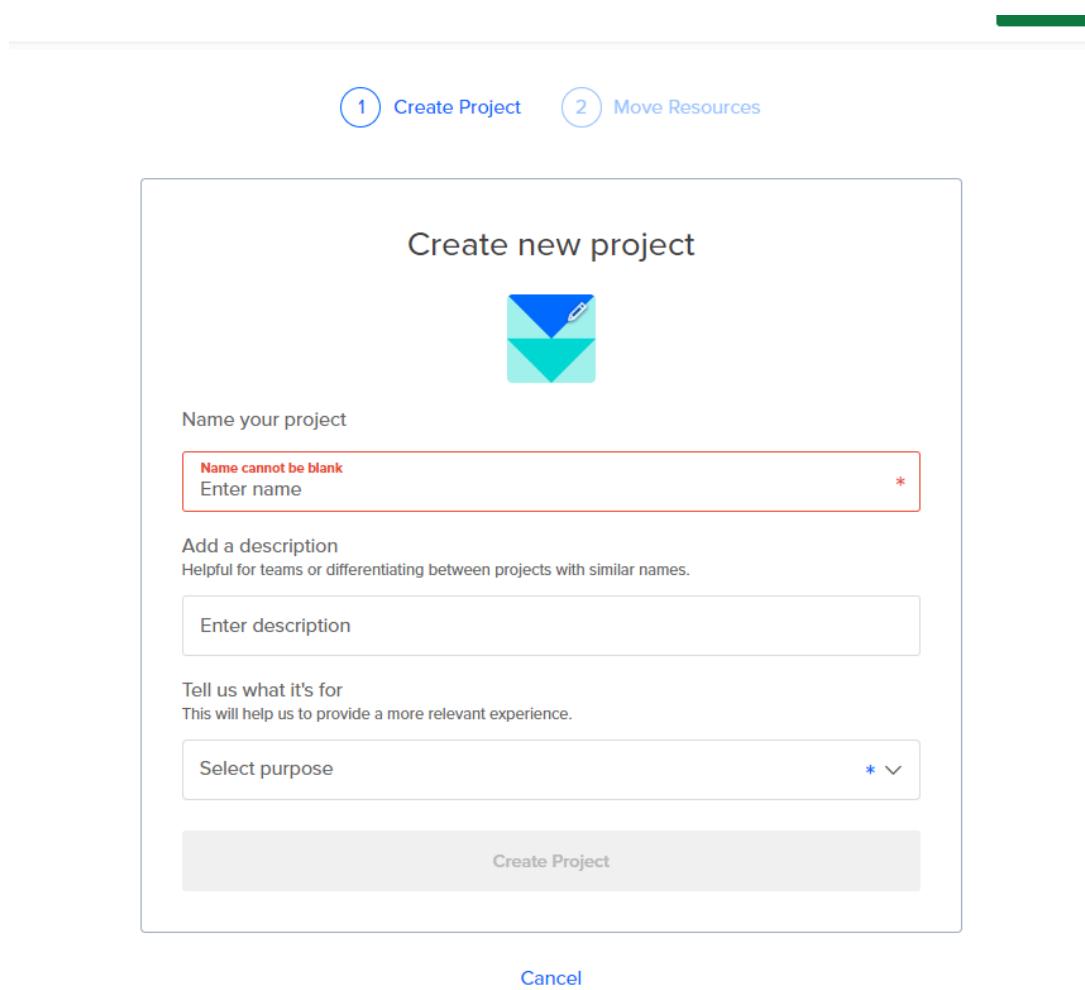
By : Yves Picicur

This guide walks you through setting up a static website on a virtual private server (VPS) using DigitalOcean. You'll create and configure a droplet, generate and add an SSH key for secure access, and set up Debian 12 as the operating system. You will then install and configure Hugo to manage your site's content, choose a theme, and build the site locally. Afterward, the guide details how to upload your site to the VPS, set up a web server (Nginx), and make the site publicly accessible. Optional steps include configuring a custom domain and enabling HTTPS for secure connections. The guide also covers content management, ensuring you can easily update and maintain your website in the future.

First step to set up the solution would be to create a droplet (the virtual private server where we would keep our static website)

This setup is after creating your digital ocean account and filling the need information, credit card and other personal info, it's very straightforward and easy so I won't include this part.

On digital ocean, you go to create new project :



The screenshot shows the 'Create new project' form on the DigitalOcean website. At the top, there are two tabs: '1 Create Project' and '2 Move Resources'. The 'Create Project' tab is active. The form has several fields:

- Name your project**: A red-bordered input field containing the placeholder "Enter name". An error message "Name cannot be blank" is displayed above the field, and a red asterisk (*) is to its right.
- Add a description**: A text area with the placeholder "Enter description". Below it, a note says "Helpful for teams or differentiating between projects with similar names."
- Tell us what it's for**: A dropdown menu labeled "Select purpose" with a red asterisk (*) to its right. The dropdown arrow is currently down.
- Create Project**: A large grey button at the bottom of the form.
- Cancel**: A blue link at the bottom left of the form.

You can fill in the information you need

Products

NEWS: Vultr Offers NVIDIA GH200 Grace Hopper Superchip: Tap Into Ultimate Power and Efficiency

Yves Picard Deploy +

Deploy New Instance

Choose Type

- Optimized Cloud Compute - Dedicated CPU
- Cloud Compute - Shared CPU**
- Cloud GPU
- Bare Metal

Virtual machines for more demanding business apps, e.g. production websites, CI/CD, video transcoding, or larger databases.

Virtual machines for apps with bursty performance, e.g. low traffic websites, blogs, CMS, dev/test environments, and small databases.

Virtual machines with fractional or full NVIDIA GPUs for AI machine learning, HPC, visual computing and VDI. Also available as Bare Metal.

Single tenant bare metal for apps with the most demanding performance or security requirements.

Choose Location

All Locations Americas Europe Australia Asia Africa

New York (NJ)	United States	Atlanta	United States	Chicago	United States	Dallas	United States
Los Angeles	United States	Mexico City	Mexico	Miami	United States	Seattle	United States
Silicon Valley	United States	Toronto	Canada	London	United Kingdom	Amsterdam	Netherlands
Frankfurt	Germany	Madrid	Spain	Manchester	United Kingdom	Paris	France
Stockholm	Sweden	Warsaw	Poland	Tokyo	Japan	Bangalore	India
Delhi NCR	India	Mumbai	India	Osaka	Japan	Seoul	South Korea
Singapore	Singapore	Tel Aviv	Israel	Sydney	Australia	Melbourne	Australia
Johannesburg	South Africa	São Paulo	Brazil	Santiago	Chile		

Servers Qty: **1** Summary: **\$6.00/month** (\$0.008/hour) Deploy Now

Additional 14.97% Total Sales Tax applicable to this order [Update Settings](#)

Products

Choose Image

Operating System Marketplace Apps Upload ISO ISO Library Backup Snapshot

AlmaLinux	Alpine Linux	Arch Linux	CentOS
Debian	Fedoroid	Fedora CoreOS	Flatcar Container Linux
FreeBSD	OpenBSD	Rocky Linux	Ubuntu
Windows Core Standard	Windows Standard	openSUSE	

Select Version: 12 v64

Select Version: Latest v64

Select Version: Latest v64

Select Version: Select Version

Choose Plan

AMD High Performance Intel High Performance High Frequency Regular Cloud Compute

Name	Cores	Memory	Storage	Bandwidth	Price
25 GB SSD	1vCPU	1 GB	25 GB SSD	1 TB	\$5/month \$0.007/hour
55 GB SSD	1vCPU	2 GB	55 GB SSD	2 TB	\$10/month \$0.014/hour
65 GB SSD	2vCPUs	2 GB	65 GB SSD	3 TB	\$15/month \$0.021/hour
80 GB SSD	2vCPUs	4 GB	80 GB SSD	3 TB	\$20/month \$0.027/hour

Servers Qty: **1** Summary: **\$6.00/month** (\$0.008/hour) Deploy Now

Additional 14.97% Total Sales Tax applicable to this order [Update Settings](#)

On this part to have access to your server, create a new pair of SSH key on your linux machine by running the command **ssh-keygen**

Your setup should look like this :

Then, you can add your SSH key by copying it

```
cat ~/.ssh/id_rsa.pub
```

```
{debian@debian:~$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAEAAQNaCzlyC2EAAAQABAAQD9yA9gPqB3Tl4e8nE2ZC3QrFAMADFjC+7xI6na9Vmz+rsxnatD9ajV1M1fa7aY365aJxRT+9RIU5eLNF0ffvGgtE5opz8Vqx2EcA/AlImqxPFPZ4KA4sSMoDUHTS3JdJNPL0e
T1Javg9WpxCLNQSuEPAJx2q1ImwPKWh79L9xKSP43aiZEE1iy3x/PsF5THinU/LFFgk2x2FhSuzMa62K0hdxQzE25LcXQXNwe4RfGdjNQnj3NIke9AUUrxxytXQah/9JcYw4a+ffodRNBToIM9QK2co9fGzxQddGgnV
St53bqy1KzP45dnVz9h3neDdNuXhb236YS8l1QsyqNf1P1/WpM5H6tRahp69zayxi1g2lGm2yqqNMk3a8Jw+iNf4auyvC8B9Gkjao0bhceT9AzqznLjGuVtE361ljCRzExBz1i1FRN+xL6GwSduFd2fZyXoeD5XhvuqAh8
9VoyzC-HXSW035WzbJxo2crYbvBvt= debian@debian
{debian@debian:~$
```

You click on New SSH Key and paste the public key in here :

Then you can modify the droplet name, you can call it anything you want but I will call it YvesDroplet.

Add public SSH key

Copy your public SSH key and paste it in the space below. For instructions on how, follow the steps on the right.

SSH key content must be a valid SSH key

SSH key content *

Name *

Add SSH Key

The result should look like this :

The point of this solution is through the whole process only use a debian 12 distribution if possible. So I am assuming you already have one setup, if not you can watch this video for example on how to set it up on your device : <https://www.youtube.com/watch?v=Gklb-l1K2FQ>

The screenshot shows a cloud provider's dashboard. At the top, there's a logo with a blue square and a white pencil icon, followed by the text "MyNewServer" and "Class project / Educational purposes / Unix Final Project". To the right is a button labeled "Move Resources". Below this, there are tabs for "Resources", "Activity", and "Settings", with "Resources" being the active tab. Under the "Resources" tab, there's a section titled "DROPLETS (1)". It lists one item: "YvesDroplet" with IP "138.197.141.218". To the right of this list are several buttons: a plus sign with a droplet icon, another plus sign with a droplet icon, a "Upsize" button, and a three-dot menu icon.

There are also plenty of other tutorials on youtube for your device.

Now let's test the VPS by connecting to it through our debian by using the command :
ssh root@yourdropletip

So for me in this case ssh root@138.197.141.218

```
{}debian@debian:~$ ssh root@138.197.141.218
The authenticity of host '138.197.141.218' (138.197.141.218) can't be established.
ED25519 key fingerprint is SHA256:cs8BgbcWoTc5MAzW03p5DKbtFd1mu+G+C1E0onJDTh4.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '138.197.141.218' (ED25519) to the list of known hosts.
Linux YvesDroplet 6.1.0-26-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.112-1 (2024-09-30) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@YvesDroplet:~#
```

You are now connected to the droplet! Anytime you want to connect to make modifications to the droplet through your terminal, you need run the same command.

For now we can work as root to avoid any access issues.

We are going to choose a theme from the hugo website <https://themes.gohugo.io> , in this case yue, a blog template <https://themes.gohugo.io/themes/hugo-theme-yue/>

Then build it locally to test it. After we make sure it is working, we are going to upload it to the VPS.

As we can see in the setup instructions it requires hugo extended, a specific version of hugo.

The screenshot shows the 'Get started' section of the Hugo documentation. It includes a 'Install' link and a code block with terminal commands for setting up a demo website. Below the code block is a note about the working demo website.

```
# Create website
git init my-website
cd my-website
# Install theme
git submodule add --depth=1 https://github.com/CyrusYip/hugo-theme-yue themes/hugo-theme-yue
git commit --message "add theme"
# Create demo content
cp --recursive themes/hugo-theme-yue/exampleSite/* .
# Preview
hugo server
```

Now we have a working demo website. The `content` directory contains the content, and `hugo.yaml` is configuration file. Feel free to play around with them.

We are going to download it directly from github to avoid any issues.

Steps to Install Hugo Extended

1. Download Hugo Extended from GitHub

Visit the [Hugo Releases Page](#) to find the latest version of Hugo Extended.

Use wget to download the appropriate version for your system. Replace <version> with the desired version number (e.g., 0.139.3) and ensure you select the **extended version** for your architecture (e.g., Linux-64bit or Linux-ARM64).

Example:

```
wget
https://github.com/gohugoio/hugo/releases/download/v0.139.3/hugo_extended_0.139.3_Linux-ARM
64.tar.gz
```

2. Extract the Downloaded File

Use tar to extract the binary:

```
tar -xvzf hugo_extended_0.139.3_Linux-ARM64.tar.gz
```

3. Move the Binary to a Directory in Your PATH

Move the hugo binary to /usr/local/bin to make it globally accessible:

```
sudo mv hugo /usr/local/bin/
```

4. Verify the Installation

Confirm that Hugo Extended is installed correctly by checking the version:

```
hugo version
```

The output should indicate that you are using the **extended version**:

```
hugo v0.139.3+extended linux/arm64 BuildDate=2024-11-29T15:36:56Z
```

5. Clean Up

Optionally, remove the downloaded .tar.gz file to save space:

```
rm hugo_extended_0.139.3_Linux-ARM64.tar.gz
```

Here is the sample of my terminal :

```
{}debian@debian:~$ tar -xvzf hugo_extended_0.139.3_linux-arm64.tar.gz
hugo
README.md
LICENSE
{}debian@debian:~$ sudo mv hugo /usr/local/bin/
[sudo] password for debian:
{}debian@debian:~$ hugo version
hugo v0.139.3-2f6864387cd31b975914e8373d4bf38bddbd47bc+extended linux/arm64 BuildDate=2024-11-29T15:36:56Z VendorInfo=gohugoio
{}debian@debian:~$
```

Now we run the commands from the hugo page to set up the theme locally and build it.

For the sake of simplicity to follow the commands, I am going to call it **my-website** .

```
# Create website
git init my-website
cd my-website

# Install theme
git submodule add --depth=1
https://github.com/CyrusYip/hugo-theme-yue
themes/hugo-theme-yue
git commit --message "add theme"

# Create demo content
cp --recursive themes/hugo-theme-yue/exampleSite/* .

# Preview
hugo server
```

```
{}debian@debian:~$ git init my-website
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/debian/my-website/.git/
{}debian@debian:~$ cd my-website
{}debian@debian:~/my-website$ git submodule add --depth=1 https://github.com/CyrusYip/hugo-theme-yue themes/hugo-theme-yue
git commit --message "add theme"
Cloning into '/home/debian/my-website/themes/hugo-theme-yue'...
remote: Enumerating objects: 116, done.
remote: Counting objects: 100% (116/116), done.
remote: Compressing objects: 100% (97/97), done.
remote: Total 116 (delta 5), reused 67 (delta 3), pack-reused 0 (from 0)
Receiving objects: 100% (116/116), 258.74 KiB | 2.49 MiB/s, done.
Resolving deltas: 100% (5/5), done.
[master (root-commit) 6ba2ef3] add theme
 2 files changed, 4 insertions(+)
 create mode 100644 .gitmodules
 create mode 160000 themes/hugo-theme-yue
{}debian@debian:~/my-website$ cp --recursive themes/hugo-theme-yue/exampleSite/* .
```

Your terminal should look like this :

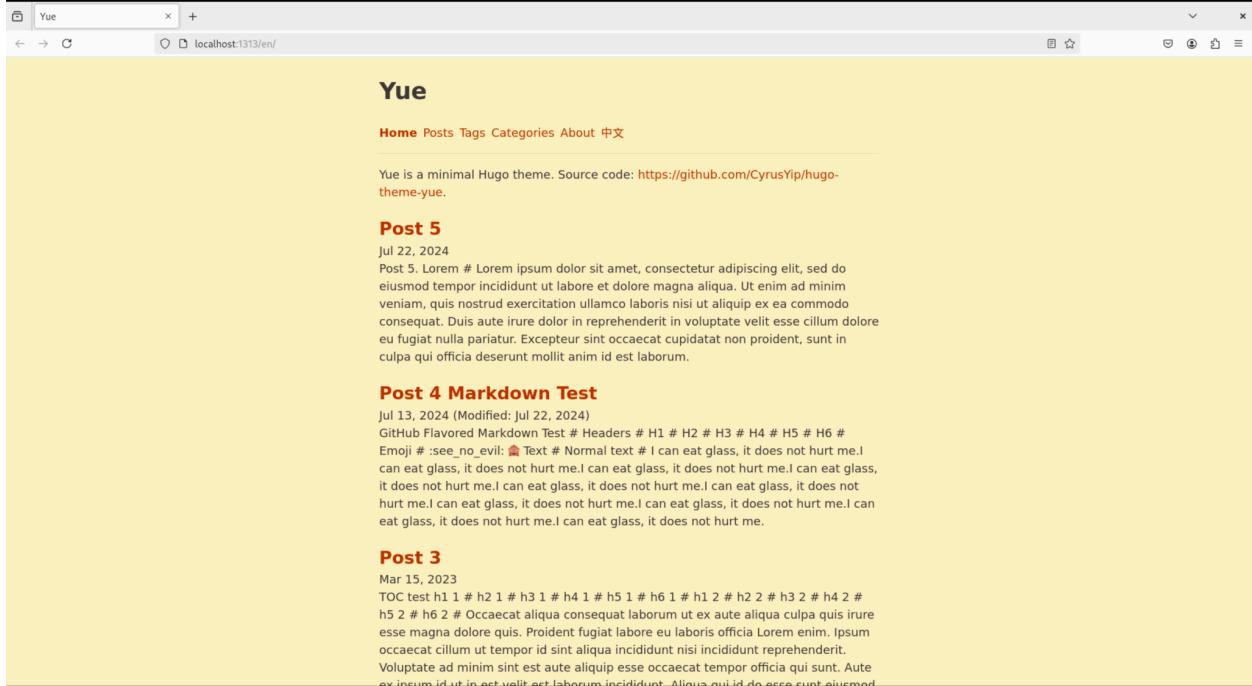
Now let's see our default theme :

```
{}debian@debian:~/my-website$ hugo server
Watching for changes in /home/debian/my-website/{content,themes}
Watching for config changes in /home/debian/my-website/hugo.yaml, /home/debian/my-website/themes/hugo-theme-yue/hugo.yaml
Start building sites ...
hugo v0.139.3-2f6864387cd31b975914e8373d4bf38bddbd47bc+extended linux/arm64 BuildDate=2024-11-29T15:36:56Z VendorInfo=gohugoio

      | EN | ZH-CN
-----+---+-----
Pages    | 38 | 36
Paginator pages | 1 | 1
Non-page files | 1 | 1
Static files  | 1 | 1
Processed images | 0 | 0
Aliases     | 3 | 2
Cleaned     | 0 | 0

Built in 68 ms
Environment: "development"
Serving pages from disk
Running in Fast Render Mode. For full rebuilds on change: hugo server --disableFastRender
Web Server is available at http://localhost:1313/ (bind address 127.0.0.1)
Press Ctrl+C to stop
■
```

By going to localhost you can preview the theme.



Now it's time to upload it to the VPS.

1. Ensure Your Droplet Is Set Up

- Ensure you can SSH into your droplet:

```
ssh root@your_droplet_ip
```

```
{}debian@debian:~$ ssh root@138.197.141.218
The authenticity of host '138.197.141.218 (138.197.141.218)' can't be established.
ED25519 key fingerprint is SHA256:cs8BgbcWoTc5MAzW03p5DKbtFd1mu+G+C1E0onJDTh4.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '138.197.141.218' (ED25519) to the list of known hosts.
Linux YvesDroplet 6.1.0-26-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.112-1 (2024-09-30) x86_64
```

- Ensure your droplet has sufficient disk space and is updated:

```
sudo apt update && sudo apt upgrade -y
```

```
root@YvesDroplet:~# sudo apt update && sudo apt upgrade -y
Get:1 file:/etc/apt/mirrors/debian.list Mirrorlist [39 B]
Get:5 file:/etc/apt/mirrors/debian-security.list Mirrorlist [27 B]
Hit:6 http://security.debian.org bookworm-security InRelease
Hit:2 http://mirrors.digitalocean.com/debian bookworm InRelease
Hit:7 https://repos-droplet.digitalocean.com/apt/droplet-agent main InRelease
Hit:3 http://mirrors.digitalocean.com/debian bookworm-updates InRelease
Hit:4 http://mirrors.digitalocean.com/debian bookworm-backports InRelease
```

2. Install and Configure a Web Server

You'll need a web server (e.g., Nginx or Apache) to serve your static Hugo site. Below is an example with Nginx.

Install Nginx

1. Install Nginx on your droplet:

```
sudo apt install nginx -y
```

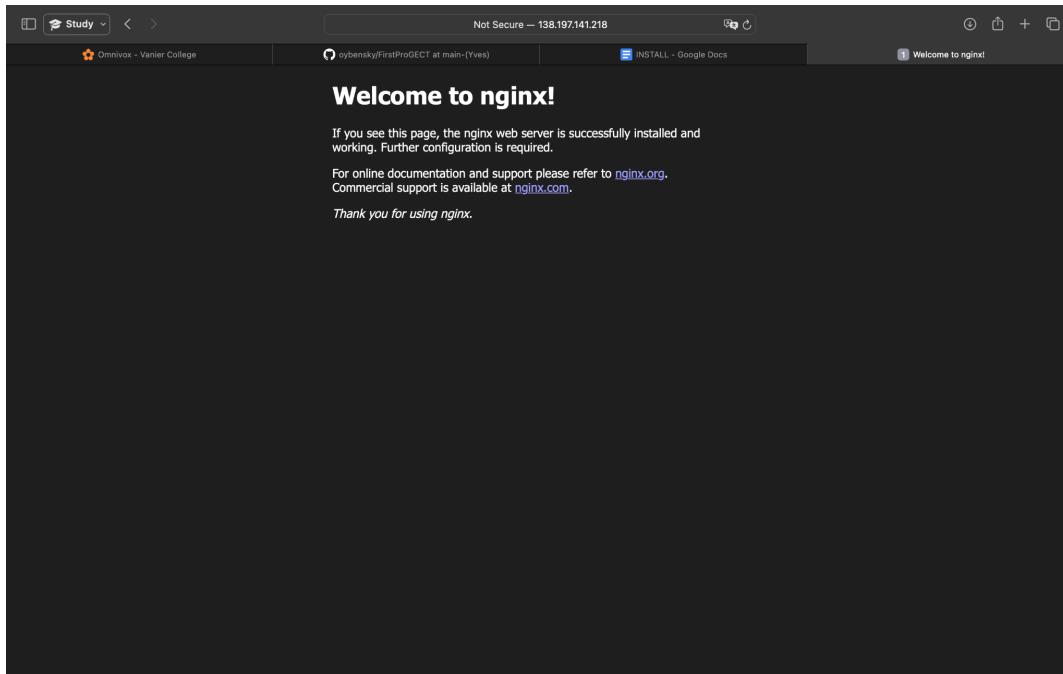
```
root@YvesDroplet:~# sudo apt install nginx -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  nginx-common
```

2. Start and enable Nginx:

```
sudo systemctl start nginx
sudo systemctl enable nginx
```

```
root@YvesDroplet:~# sudo systemctl start nginx
sudo systemctl enable nginx
Synchronizing state of nginx.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable nginx
```

3. Test Nginx by visiting `http://your_droplet_ip`. You should see the default Nginx page.



3. Build Your Hugo Site Locally

1. Navigate to your Hugo project directory on your local machine:
cd ~/my-website

2. Build the site:

Hugo

```
{}debian@debian:~/my-website$ hugo
Start building sites ...
hugo v0.139.3-2f6864387cd31b975914e8373d4bf38bddbd47bc+extended linux/arm64 BuildDate=2024-11-29T15:36:56Z VendorInfo=gohugoio
```

	EN	ZH-CN
Pages	38	36
Paginator pages	1	1
Non-page files	1	1
Static files	1	1
Processed images	0	0
Aliases	3	2
Cleaned	0	0

This will generate the static files in the public/ directory.

4. Upload the Static Files to the Droplet

1. Use scp to transfer the public/ directory to the droplet:
scp -r public/* root@your_droplet_ip:/var/www/html/

Replace `your_droplet_ip` with the IP of your droplet.

2. Verify that the files were transferred to /var/www/html:

ssh root@your_droplet_ip

ls /var/www/html

```
root@YvesDroplet:~/# ls /var/www/html
css en favicon.ico index.html index.nginx-debian.html robots.txt sitemap.xml zh-cn
```

5. Configure Nginx to Serve the Website

1. Open the Nginx configuration file:

```
sudo nano /etc/nginx/sites-available/default
```

```
root@YvesDroplet:~# sudo nano /etc/nginx/sites-available/default
```

2. Update the root directive to point to /var/www/html, and configure the server_name directive:

```
server {  
    listen 80;  
    server_name your_domain_or_ip;  
    root /var/www/html;  
    index index.html;  
    location / {  
        try_files $uri $uri/ =404;  
    }  
}
```

```

# Add index.php to the list if you are using PHP
index index.html;

server_name 138.197.141.218;

location / {
    # First attempt to serve request as file, then
    # as directory, then fall back to displaying a 404.
    try_files $uri $uri/ =404;
}

server {
    listen 80;
    #listen [::]:80 default_server;

    # SSL configuration
    #
    # listen 443 ssl default_server;
    # listen [::]:443 ssl default_server;
    #
    # Note: You should disable gzip for SSL traffic.
    # See: https://bugs.debian.org/773332
    #
    # Read up on ssl_ciphers to ensure a secure configuration.
    # See: https://bugs.debian.org/765782
    #
    # Self signed certs generated by the ssl-cert package
    # Don't use them in a production server!
    #
    # include snippets/snakeoil.conf;

    root /var/www/html;■
}

```

3. Save and exit (Ctrl+O, Enter, Ctrl+X).

4. Restart Nginx to apply the changes:

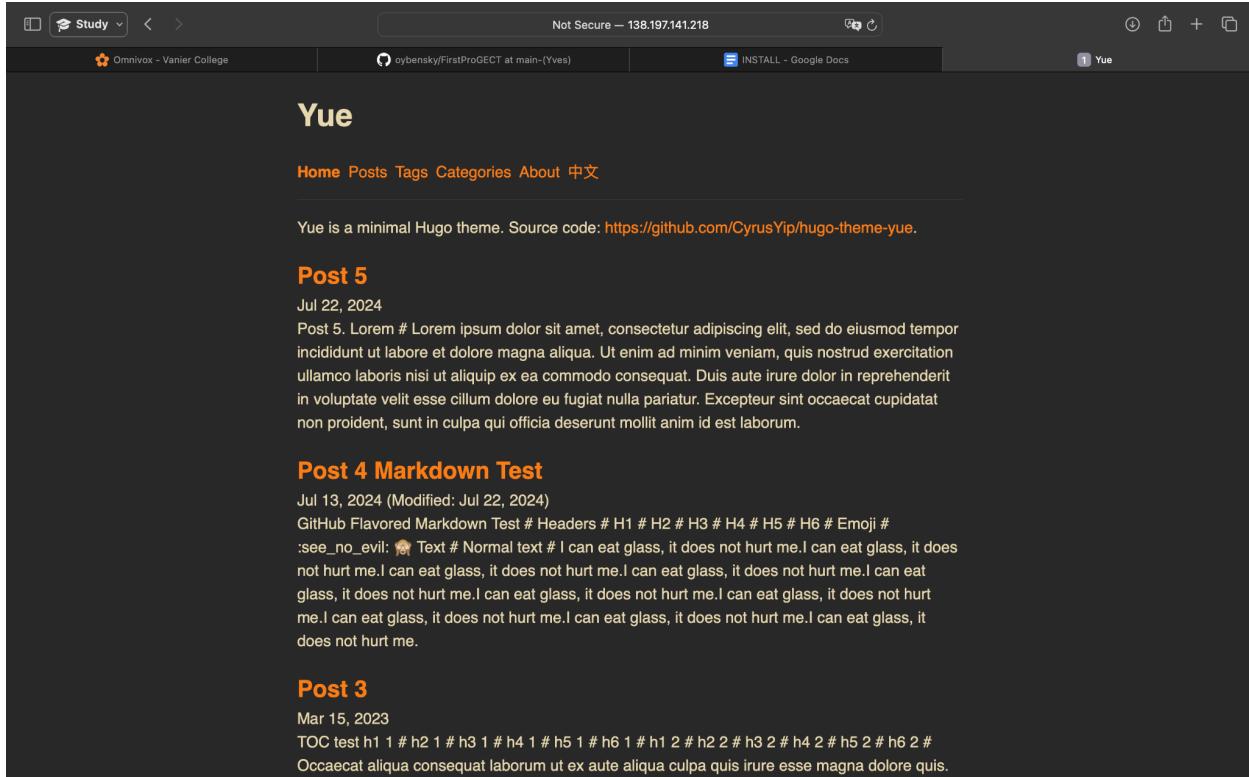
sudo systemctl restart nginx

root@YvesDroplet:~# sudo systemctl restart nginx

6. Test Your Website

1. Visit your droplet's IP address in a browser:

`http://your_droplet_ip`



You should see your Hugo site.

Since this part is optional, to approach our final goal faster we are going to skip the optional parts, but if you want you can complete the optional part.

7. (Optional) Set Up a Custom Domain

If you want to use a custom domain, point your domain to your droplet's IP address:

1. Go to your domain registrar and create an **A record** pointing to your droplet's IP.

2. Update the `server_name` in the Nginx configuration to your domain:

`server_name yourdomain.com www.yourdomain.com;`

3. Restart Nginx:

`sudo systemctl restart nginx`

4. Test by visiting your domain in a browser.

8. (Optional) Enable HTTPS with Let's Encrypt

1. Install Certbot:

`sudo apt install certbot python3-certbot-nginx -y`

2. Obtain an SSL certificate:

```
sudo certbot --nginx -d yourdomain.com -d www.yourdomain.com
```

3. Test the renewal process:

```
sudo certbot renew --dry-run
```

9. Monitor and Maintain Your Website

- Check Nginx status:

```
sudo systemctl status nginx
```

- View server logs for troubleshooting:

```
sudo tail -f /var/log/nginx/access.log
```

```
sudo tail -f /var/log/nginx/error.log
```

Summary of Key Commands

Local Machine:

```
cd ~/my-website
```

```
hugo
```

```
scp -r public/* root@your_droplet_ip:/var/www/html/
```

On the Droplet:

```
sudo apt update && sudo apt upgrade -y
```

```
sudo apt install nginx -y
```

```
sudo systemctl start nginx
```

```
sudo nano /etc/nginx/sites-available/default
```

```
sudo systemctl restart nginx
```

Test the site:

```
http://your_droplet_ip
```

To modify the content of your Hugo website, follow these steps:

1. Modify Content Locally

Hugo organizes website content into various sections like posts, pages, or custom content types. Here's how you can update or add content:

Step 1: Navigate to Your Hugo Project Directory

On your local machine, go to your Hugo project directory:

```
cd ~/my-website
```

Step 2: Edit Existing Content

1. Locate your content files (usually in content/):

```
ls content/
```

```
{}debian@debian:~/my-website$ ls content/  
en zh-CN
```

You'll see subdirectories like posts, about, etc.

2. Open the file you want to edit:
nano content/posts/your-post.md
3. Make the desired changes (e.g., update text, title, or metadata).
4. Save and exit (Ctrl+O, Enter, Ctrl+X).

In this part, most of the files are modifiable, I won't go in depth because there are too many, but the following terminal screenshots show what you can do.

```
GNU nano 7.2                                post-5.md *  
---  
date: 2024-07-22T00:00:00+08:00  
tags:  
- red  
title: This is Yves Test Website  
slug: post-5  
---  
  
Post 5.  
  
## Lorem  
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex  
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex  
  
en:  
  contentDir: content/en  
  languageCode: en-US  
  languageDirection: ltr  
  languageName: English  
  title: We love Tassia's class! # Shown in home page  
  weight: 10  
  menus:  
    main:  
      - name: Home  
        pageRef: /
```

```
{}debian@debian:~/my-website$ ls
content hugo.yaml public resources themes
{}debian@debian:~/my-website$ cd content/
{}debian@debian:~/my-website/content$ ls
en zh-CN
{}debian@debian:~/my-website/content$ cd en/
{}debian@debian:~/my-website/content/en$ ls
about.md categories _index.md posts tags
{}debian@debian:~/my-website/content/en$ cd posts/
{}debian@debian:~/my-website/content/en/posts$ ls
_index.md markdown-syntax.md placeholder-text.md post-1.md post-2.md post-3 post-4.md post-5.md
{}debian@debian:~/my-website/content/en/posts$ nano post-5.md
{}debian@debian:~/my-website/content/en/posts$ cd ..
{}debian@debian:~/my-website/content/en$ cd ..
{}debian@debian:~/my-website/content$ cd ..
{}debian@debian:~/my-website$ nano hugo.yaml
{}debian@debian:~/my-website$ █
```

Step 3: Add New Content

1. Create a new content file using Hugo's new command:

`hugo new content/posts/my-new-post.md`

This generates a new file with front matter (metadata) based on your theme's default template.

2. Edit the new file:

`nano content/posts/my-new-post.md`

Add your content in Markdown format.

2. Preview Changes Locally

1. Run the Hugo development server:

`hugo server`

```
{}debian@debian:~/my-website$ hugo server
Watching for changes in /home/debian/my-website/{content,themes}
Watching for config changes in /home/debian/my-website/hugo.yaml, /home/debian/my-website/themes/hugo-theme-yue/hugo.yaml
Start building sites ...
hugo v0.139.3-2f6864387cd31b975914e8373d4bf38bddbd47bc+extended linux/arm64 BuildDate=2024-11-29T15:36:56Z VendorInfo=gohugoio
```

	EN	ZH-CN
Pages	38	36
Paginator pages	1	1
Non-page files	1	1
Static files	1	1
Processed images	0	0
Aliases	3	2
Cleaned	0	0

```
Built in 71 ms
Environment: "development"
Serving pages from disk
Running in Fast Render Mode. For full rebuilds on change: hugo server --disableFastRender
Web Server is available at http://localhost:1313/ (bind address 127.0.0.1)
Press Ctrl+C to stop
█
```

2. Visit the preview URL (e.g., `http://localhost:1313/`) in your browser to see your changes.

3. Rebuild the Site

Once you're satisfied with the changes, rebuild the site:

`hugo`

```
^C{}debian@debian:~/my-website$ hugo
Start building sites ...
hugo v0.139.3-2f6864387cd31b975914e8373d4bf38bddbd47bc+extended linux/arm64 BuildDate=2024-11-29T15:36:56Z VendorInfo=gohugoio
```

	EN	ZH-CN
Pages	38	36
Paginator pages	1	1
Non-page files	1	1
Static files	1	1
Processed images	0	0
Aliases	3	2
Cleaned	0	0

Total in 84 ms

This regenerates the static files in the public/ directory.

4. Upload the Updated Site to the Droplet

1. Remove the old files on the droplet (to avoid stale content):

```
ssh root@138.197.141.218
```

```
sudo rm -rf /var/www/html/*
```

```
{}debian@debian:~$ ssh root@138.197.141.218
Linux YvesDroplet 6.1.0-26-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.112-1 (2024-09-30) x86_64
```

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

Last login: Tue Dec 10 22:40:14 2024 from 24.200.162.161

```
root@YvesDroplet:~# sudo rm -rf /var/www/html/*
root@YvesDroplet:~#
```

2. Re-upload the updated public/ directory:

```
scp -r public/* root@138.197.141.218:/var/www/html/
```

```
({})debian@debian:~/my-website$ scp -r public/* root@138.197.141.218:/var/www/html/
style.css
style.min.90c0c98a5ecc51fb23066c6996d553c1d7a91daade21b9c038f9421b958c1c88d.css
index.html
```

```
100% 13KB 326.8KB/s 00:00
100% 7134 222.9KB/s 00:00
100% 3267 98.1KB/s 00:00
```

5. Restart the Web Server (If Necessary)

Restart Nginx to ensure it serves the updated content:

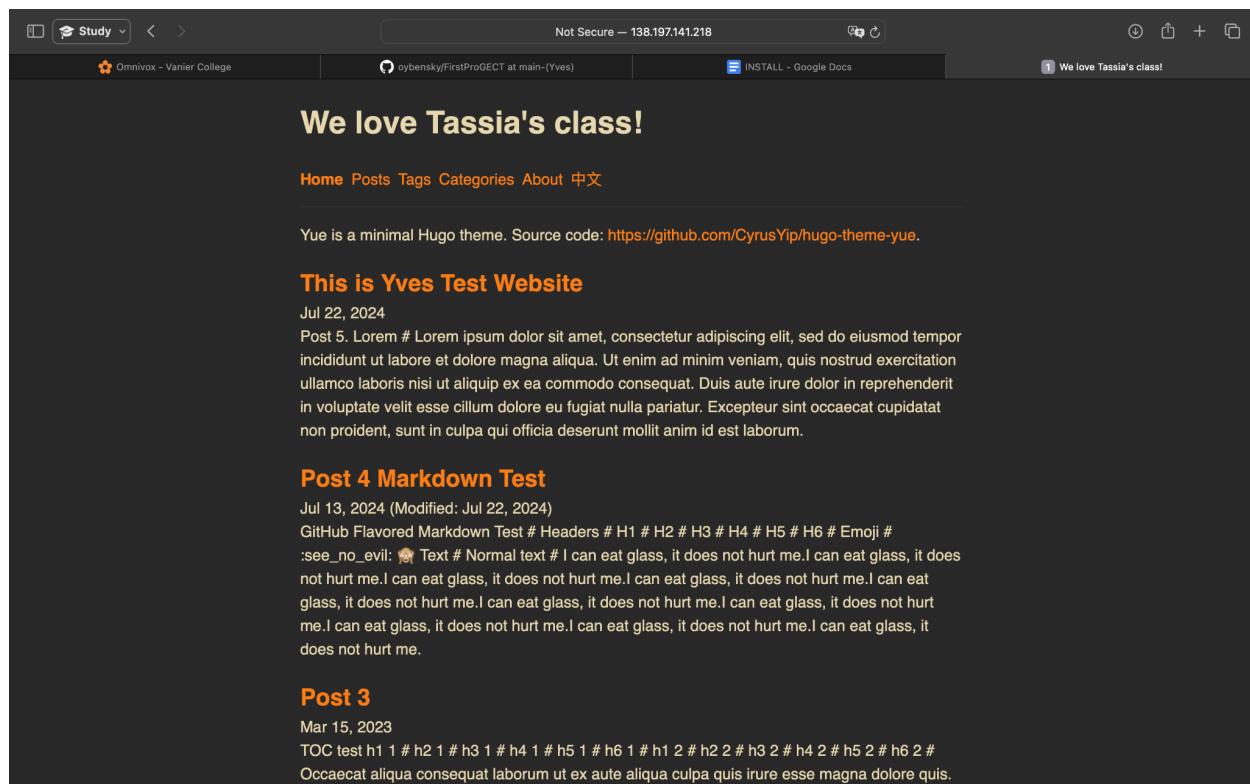
```
ssh root@138.197.141.218
```

```
sudo systemctl restart nginx
```

6. Verify Your Changes

Visit your site in a browser:

- For the droplet: <http://138.197.141.218>
- For a custom domain (if configured): <http://yourdomain.com>



7. Workflow for Future Edits

To streamline your workflow:

1. Modify content locally (content/ directory).

2. Preview using:

hugo server

3. Build the site:

hugo

4. Upload files to the droplet using:

scp -r public/* root@138.197.141.218:/var/www/html/

If you see you getting redirected towards another URL follow these steps ;

The redirection to <https://yue.cyrusyip.org/en/> happens because the baseURL in the Hugo configuration file (hugo.yaml or config.toml) is still set to the default demo site URL for the Hugo theme **Yue**. You need to update the baseURL to your own IP address or domain name.

Steps to Fix the Redirection

1. **Open Your Hugo Configuration File**

The Hugo configuration file is usually located in the root of your project directory. Depending on your setup, it might be named:

- config.toml
- config.yaml
- config.json

For example:

nano ~/my-website/config.yaml

2. **Update the baseURL**

Locate the baseURL setting and change it to match your droplet's IP or your custom domain.

If using an IP address:

baseURL: <http://138.197.141.218/>

If using a custom domain:

baseURL: <https://yourdomain.com/>

```
# Documentation: https://gohugo.io/getting-started/configuration
# Don't remove _merge, which imports default config
theme: hugo-theme-yue
baseURL: http://138.197.141.218/ # Change this to your real domain name, e.g. https://my-cool-domain.org/
defaultContentLanguage: en
defaultContentLanguageInSubdir: true
enableGitInfo: false # Set lastmod from git history, this may increase build time
enableEmoji: true # Emoji list: https://gohugo.io/quick-reference/emojis/
enableRobotsTXT: true
```

3. Rebuild the Hugo Site

Rebuild your Hugo site to apply the changes:

```
hugo
```

4. Re-upload the Static Files to Your Droplet

Replace the existing files in /var/www/html on your droplet:

```
scp -r public/* root@138.197.141.218:/var/www/html/
```

5. Test the Website

- Open a browser and visit:
<http://138.197.141.218>
- If using HTTPS and a domain:
<https://yourdomain.com>

Verify the Fix

If the issue persists:

1. Clear your browser cache or test in an incognito/private window.
2. Ensure no old Hugo-generated files are cached:

```
rm -rf /var/www/html/*
```

Then re-upload the fresh public/ directory files.

3. Restart Nginx to ensure it serves the updated files:

```
sudo systemctl restart nginx
```

Note to reader : I will keep the server running until the 27th december for showcasing purposes.