ICT 171 Assignment 2

Cloud Server Project

Name: Muhammad Amash Khan

ID: 35378947

IP address: http:// 135.235.170.99/

Domain: www.cybersectip.online

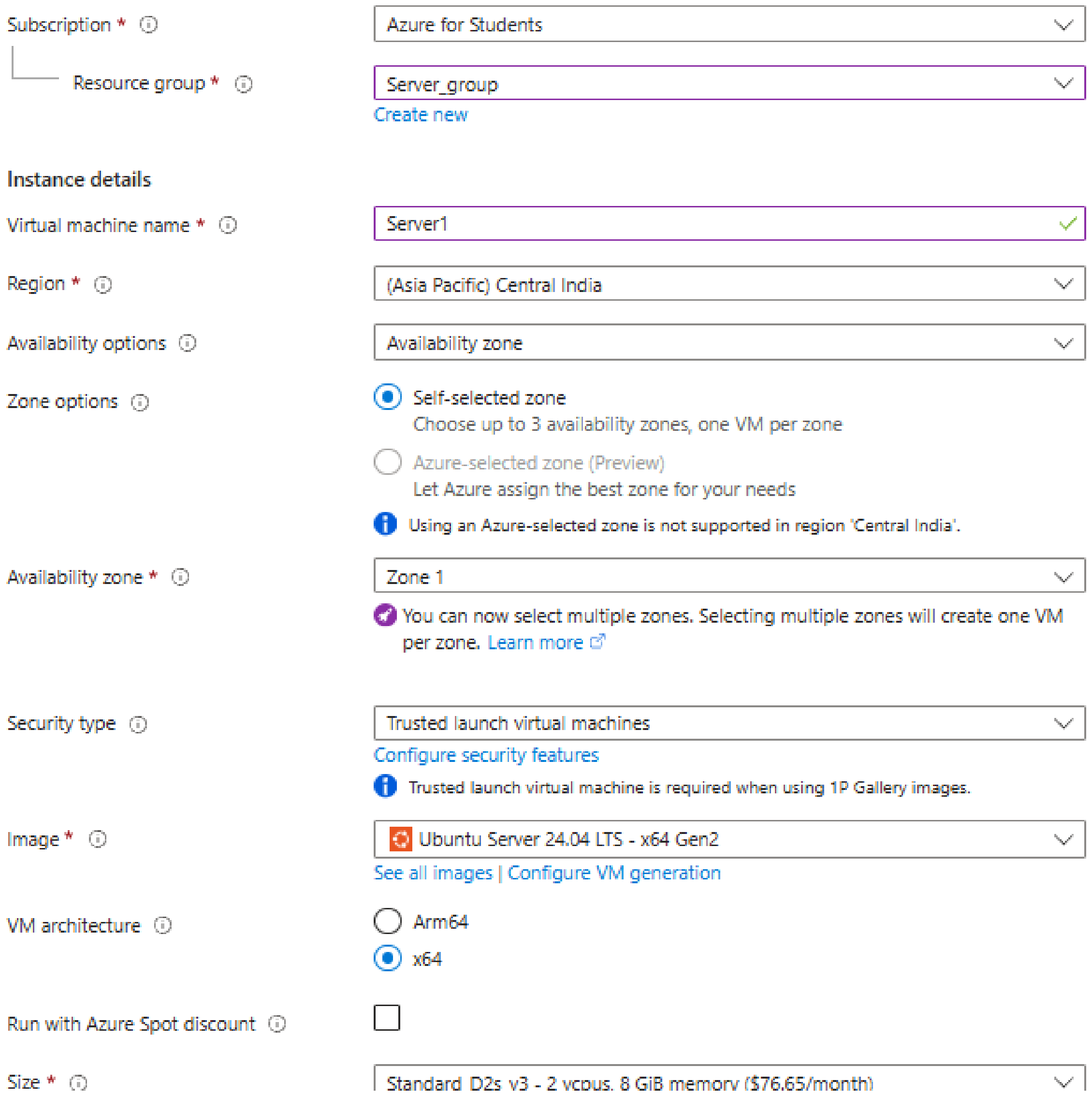
Table of Contents

Note: Everything up until the TLS Certificate was done during the first assignment, hence had to document it all over again.

Setting up the VM on Azure

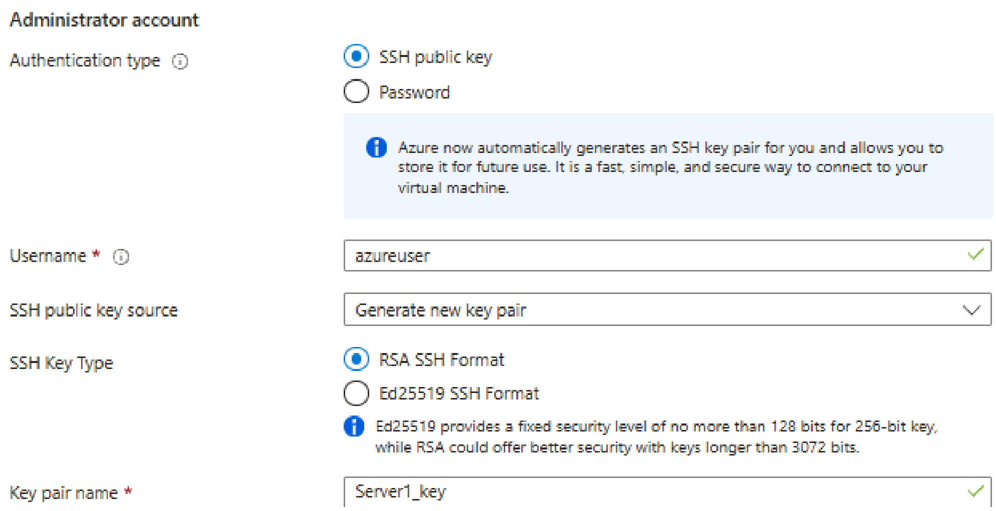
1. Log in or sign up to Microsoft Azure.
2. Once Logged in, the dashboard will appear.
3. Click on **virtual machines.**
4. Click on **create**.
5. Select the first option; “**Azure Virtual Machine**”.
6. Select the **subscription** (Azure Students in my case). Select the resource **group**. Fill in the **name** for the VM. Select **Region** (Default, closest to you), and finally the **image**, Ubuntu 24.04 for my case with the default size to

save on costs.



Subscription *	Azure for Students
Resource group *	Server_group Create new
Instance details	
Virtual machine name *	Server1 ✓
Region *	(Asia Pacific) Central India
Availability options	Availability zone
Zone options	● Self-selected zone Choose up to 3 availability zones, one VM per zone ○ Azure-selected zone (Preview) Let Azure assign the best zone for your needs Using an Azure-selected zone is not supported in region 'Central India'.
Availability zone *	Zone 1 You can now select multiple zones. Selecting multiple zones will create one VM per zone. Learn more
Security type	Trusted launch virtual machines Configure security features Trusted launch virtual machine is required when using 1P Gallery images.
Image *	Ubuntu Server 24.04 LTS - x64 Gen2 See all images | Configure VM generation
VM architecture	○ Arm64 ● x64
Run with Azure Spot discount	□
Size *	Standard D2s v3 - 2 vcpus, 8 GiB memory ($76.65/month)

7. Use SSH for authentication



Administrator account

Authentication type ⓘ
● SSH public key
○ Password

Azure now automatically generates an SSH key pair for you and allows you to store it for future use. It is a fast, simple, and secure way to connect to your virtual machine.

Username * ⓘ: azureuser ✓

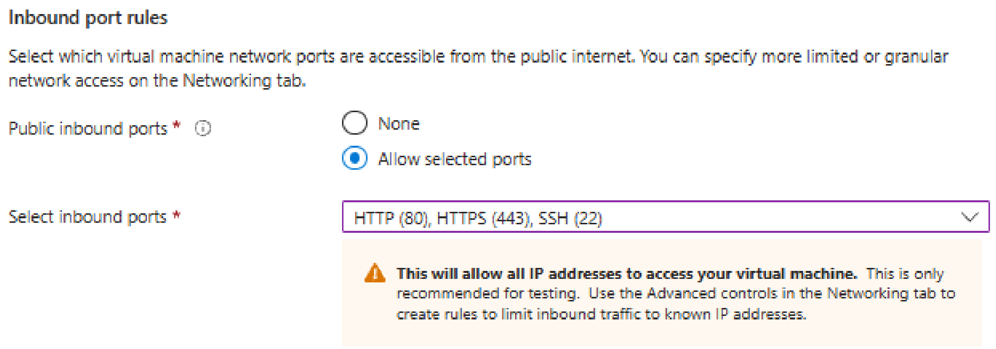
SSH public key source: Generate new key pair

SSH Key Type
● RSA SSH Format
○ Ed25519 SSH Format

Ed25519 provides a fixed security level of no more than 128 bits for 256-bit key, while RSA could offer better security with keys longer than 3072 bits.

Key pair name *: Server1_key ✓

8. Allow HTTP 80, HTTPS 443 and SSH 22 to ensure the site runs smooth



Inbound port rules

Select which virtual machine network ports are accessible from the public internet. You can specify more limited or granular network access on the Networking tab.

Public inbound ports * ⓘ
○ None
● Allow selected ports

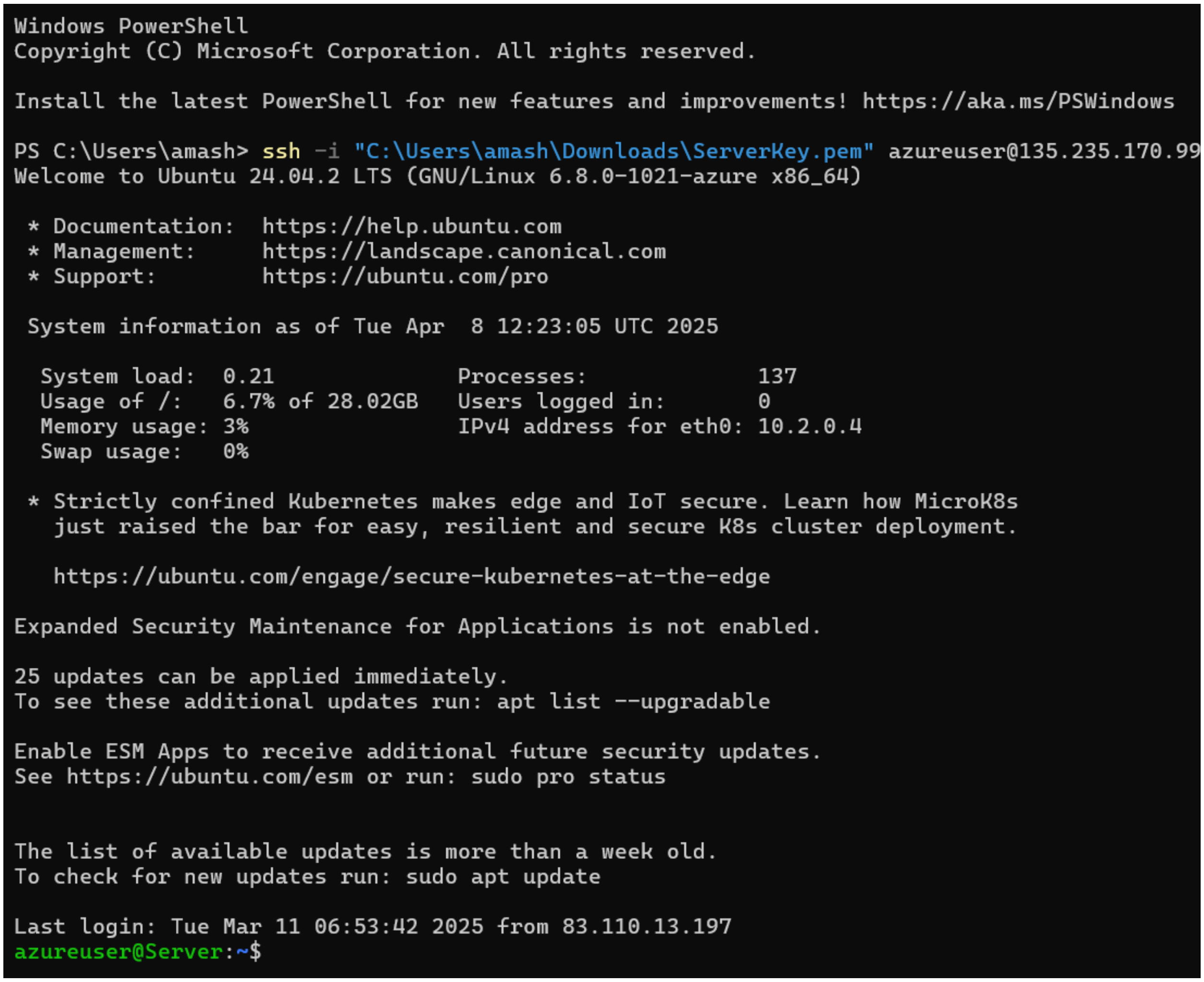
Select inbound ports *: HTTP (80), HTTPS (443), SSH (22)

This will allow all IP addresses to access your virtual machine. This is only recommended for testing. Use the Advanced controls in the Networking tab to create rules to limit inbound traffic to known IP addresses.

9. Click on **review+create** and then **create**.

Log in and Install Nginx Server

1. Copy the path of the Key for your Server
2. Post it in the Native SSH portion of the connect tab in Azure
3. It will give a code like: **ssh -i "C:\Users\amash\Downloads\ServerKey.pem" azureuser@135.235.170.99** (My key name is different as I did everything up to the TLS certificate part in Assignment 1 and did not know I had to document it then).



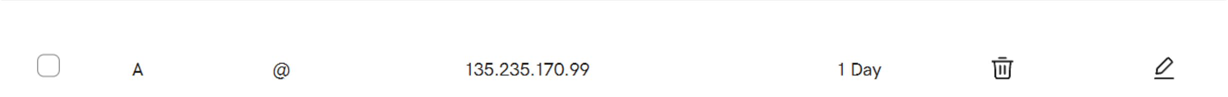
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\amash> ssh -i "C:\Users\amash\Downloads\ServerKey.pem" azureuser@135.235.170.99
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-1021-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Tue Apr  8 12:23:05 UTC 2025

  System load:  0.21              Processes:             137
  Usage of /:   6.7% of 28.02GB   Users logged in:       0
  Memory usage: 3%                IPv4 address for eth0: 10.2.0.4
  Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

25 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Tue Mar 11 06:53:42 2025 from 83.110.13.197
azureuser@Server:~$
```

4. Now we can install Nginx with the following commands:
   - **Sudo apt update**
   - **Sudo apt install nginx -y**
   - **Sudo systemctl start nginx**
   - **Sudo systemctl enable nginx**

Assigning my DNS to my IP

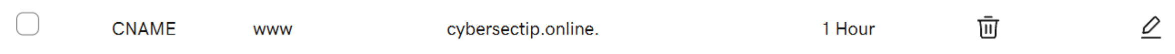
These are assuming you purchase your domain from Go daddy like I have.

1. After purchasing you Domain Name from go daddy, go to the dashboard.
2. Click on **Domain**
3. Click on **DNS**
4. Edit the **Type A value**, leave the name as **@** and put your **IP address** into the value field. Leave the TTL to **1 hour**.



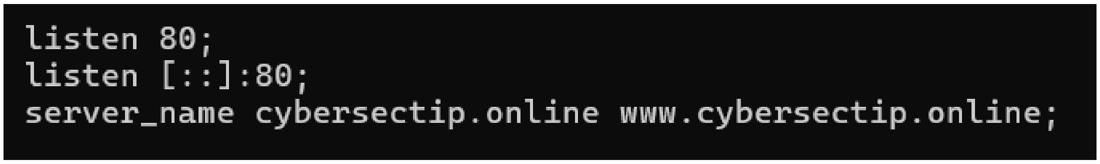
□	A	@	135.235.170.99	1 Day

5. In DNS still, find or add **CNAME** record, make the name "**www**" and the value as your **DNS, cybersectip.online** in my case.



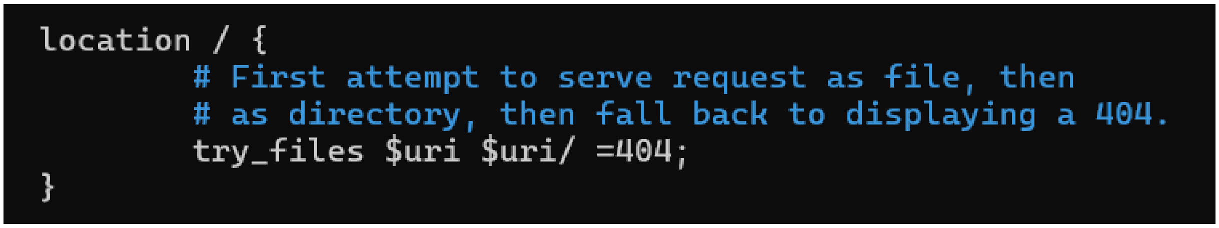
□	CNAME	www	cybersectip.online.	1 Hour

6. Now go to your server, open your default config file by typing: **sudo nano /etc/nginx/sites-available/default**
7. And enter this code into the **second server block**:



```
listen 80;
listen [::]:80;
server_name cybersectip.online www.cybersectip.online;
```

8. And this into the **first server block** which is used to set up HTTPS:



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

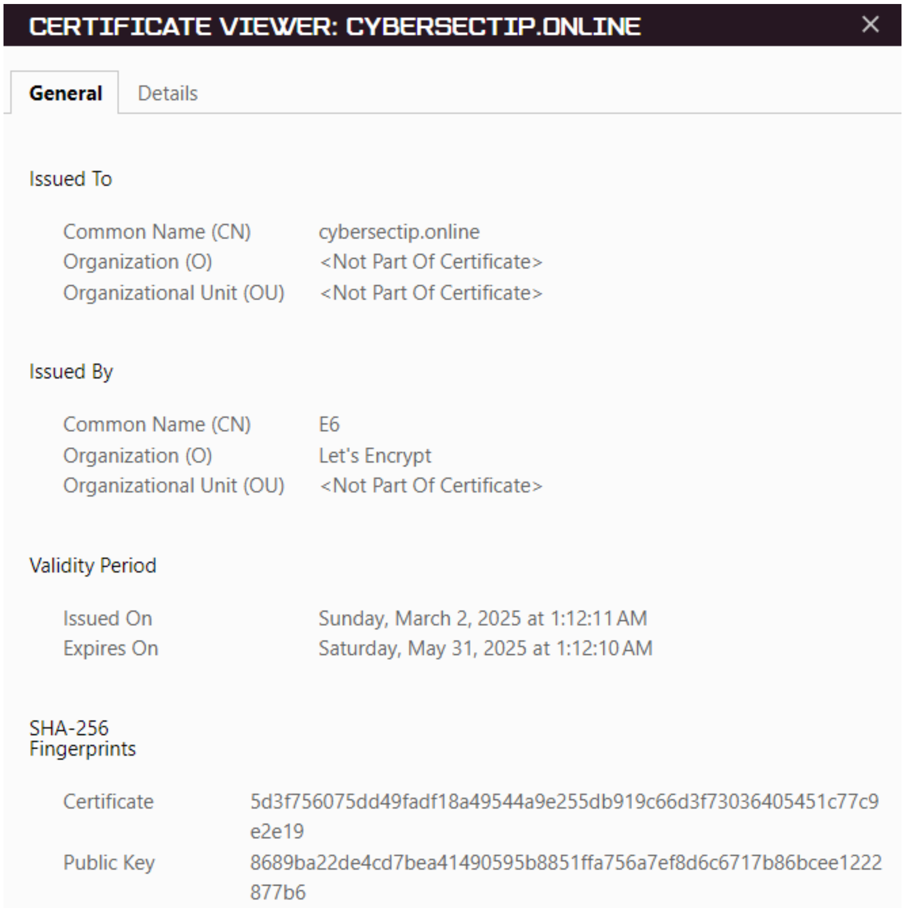
9. Now test the nginx configuration by typing: **sudo nginx -t**
10. Now reload the nginx server to apply the changes by typing: **sudo systemctl reload nginx**

Getting a TLS certificate for my website

1. Install Certbot and nginx plug in by typing : **sudo apt install certbot python3-certbot-nginx -y**
2. Run Certbot by typing: **sudo certbot --nginx -d cybersectip.online -d www.cybersectip.online**

This will give the site the TLS certificate, just replace “cybersectip.online” with your DNS.

Here is the certificate:



Issued To

Common Name (CN) cybersectip.online
Organization (O) <Not Part Of Certificate>
Organizational Unit (OU) <Not Part Of Certificate>

Issued By

Common Name (CN) E6
Organization (O) Let's Encrypt
Organizational Unit (OU) <Not Part Of Certificate>

Validity Period

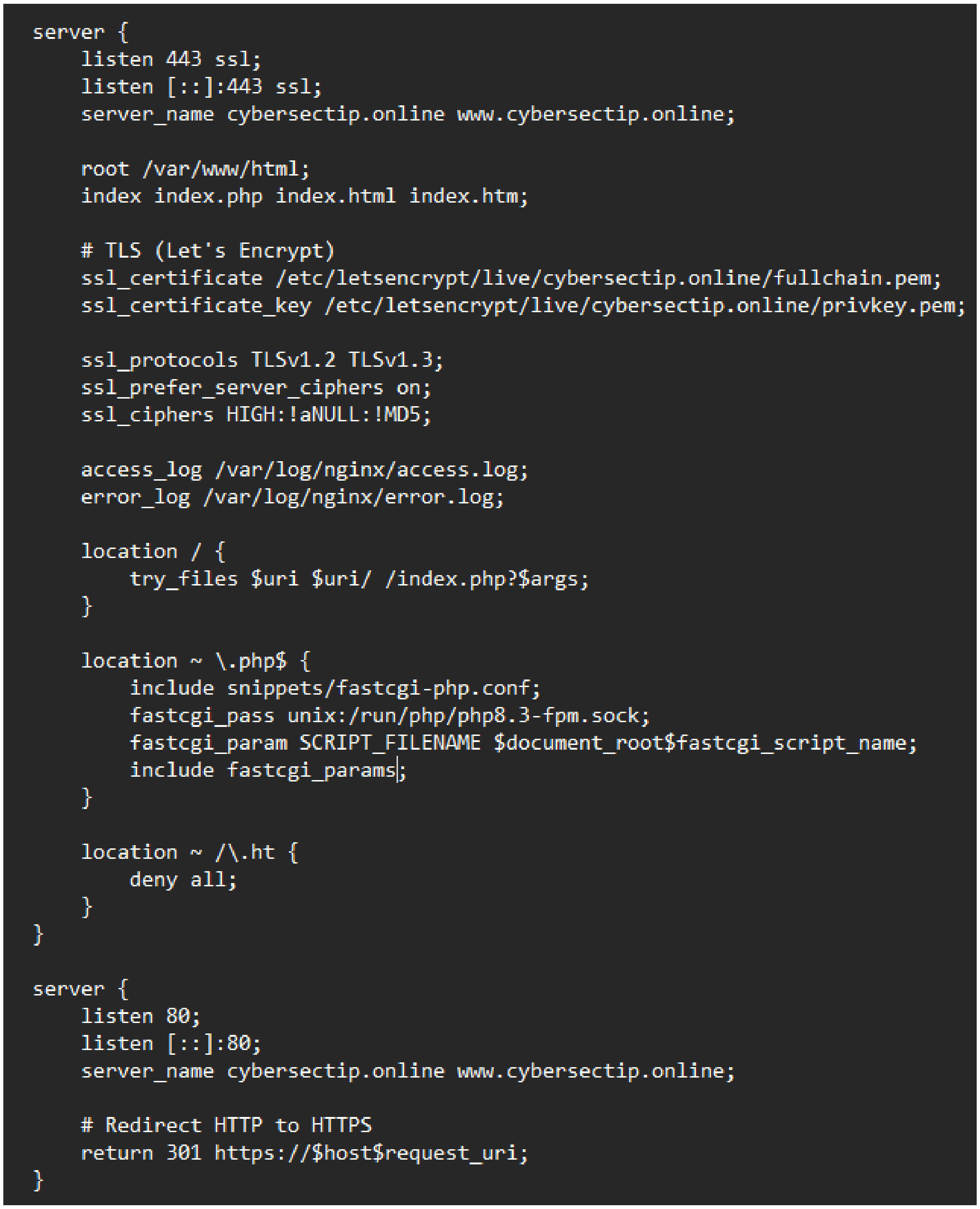
Issued On Sunday, March 2, 2025 at 1:12:11 AM
Expires On Saturday, May 31, 2025 at 1:12:10 AM

SHA-256 Fingerprints

Certificate 5d3f756075dd49fadf18a49544a9e255db919c66d3f73036405451c77c9e2e19
Public Key 8689ba22de4cd7bea41490595b8851ffa756a7ef8d6c6717b86bcee1222877b6

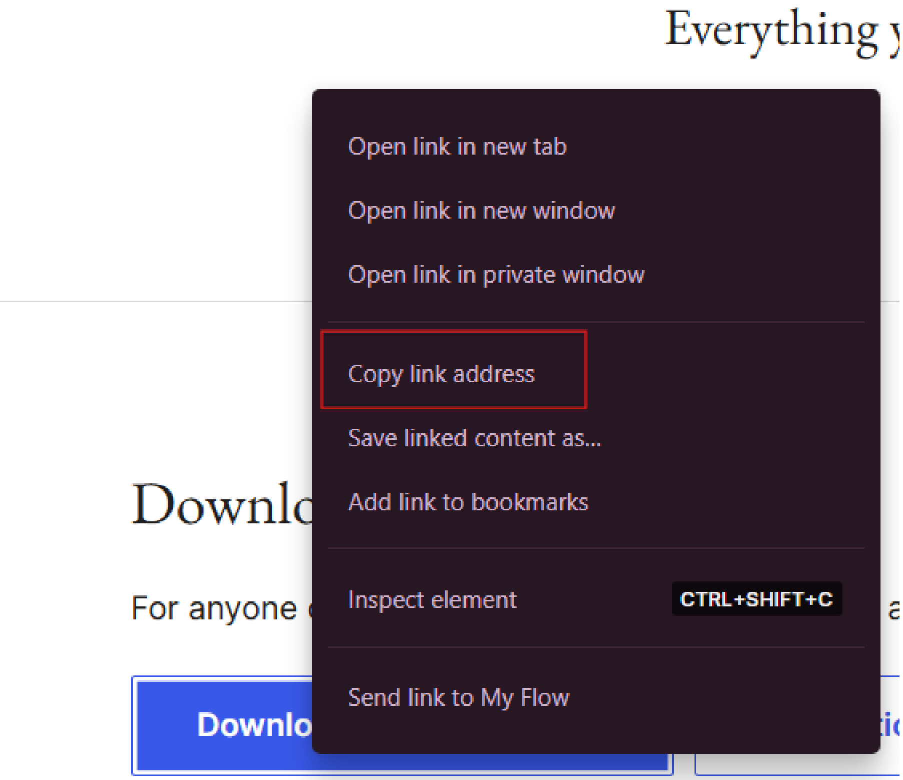
Installing and setting up WordPress on Nginx

1. Update the package list: **sudo apt update**
2. Install the required packages: **sudo apt install php-fpm php-mysql mysql-server unzip**
3. Go into the php directory to check the php version: **cd /var/run/php/**
4. Go into the sites available directory to edit the default file: **cd /etc/nginx/sites-available/**
5. Update the default file to look like this (Mine includes the settings for the TLS and my DNS, you can copy paste this, just make changes according to your DNS, TLS certificates, and PHP version:



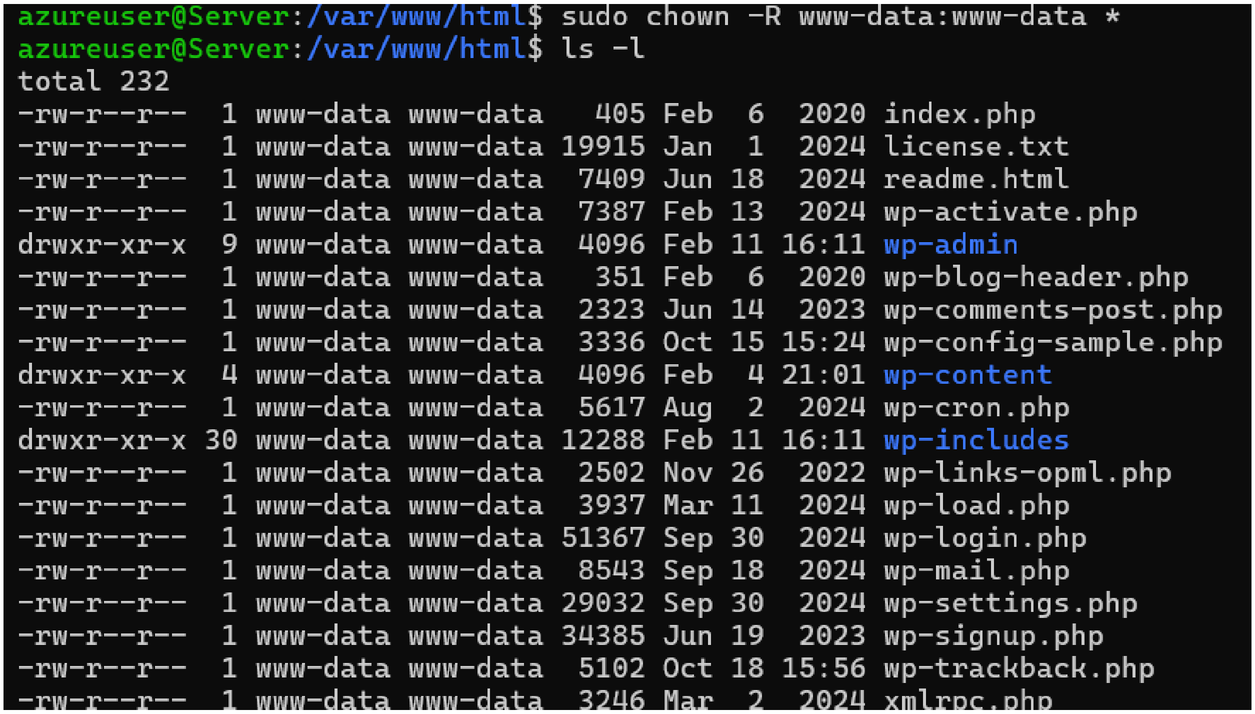
```
server {
    listen 443 ssl;
    listen [::]:443 ssl;
    server_name cybersectip.online www.cybersectip.online;

    root /var/www/html;
    index index.php index.html index.htm;

    # TLS (Let's Encrypt)
    ssl_certificate /etc/letsencrypt/live/cybersectip.online/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/cybersectip.online/privkey.pem;

    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_prefer_server_ciphers on;
    ssl_ciphers HIGH:!aNULL:!MD5;

    access_log /var/log/nginx/access.log;
    error_log /var/log/nginx/error.log;

    location / {
        try_files $uri $uri/ /index.php?$args;
    }

    location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/run/php/php8.3-fpm.sock;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        include fastcgi_params;
    }

    location ~ /\.ht {
        deny all;
    }
}

server {
    listen 80;
    listen [::]:80;
    server_name cybersectip.online www.cybersectip.online;

    # Redirect HTTP to HTTPS
    return 301 https://$host$request_uri;
}
```

6. Now restart the system using: **sudo systemctl restart nginx.service**
7. Now lets go to the html directory so we can replace the default loading page: **cd /var/www/html/**
8. After removing, lets go to www.wordpress.org to install word press, click on get WordPress. Then copy the link address of the download button:



9. Now type: **sudo wget** followed by the link you copied to start the download process.
10. Now unzip the latest.zip file created by typing: **sudo unzip latest.zip.**
11. Remove the zip archive: **sudo rm latest.zip**
12. To have wordpress accesible from the homepage of the server move the files: **sudo mv wordpress/* .**
13. Change owners of the files to the nginx user: **sudo chown -R www-data:www-data ***

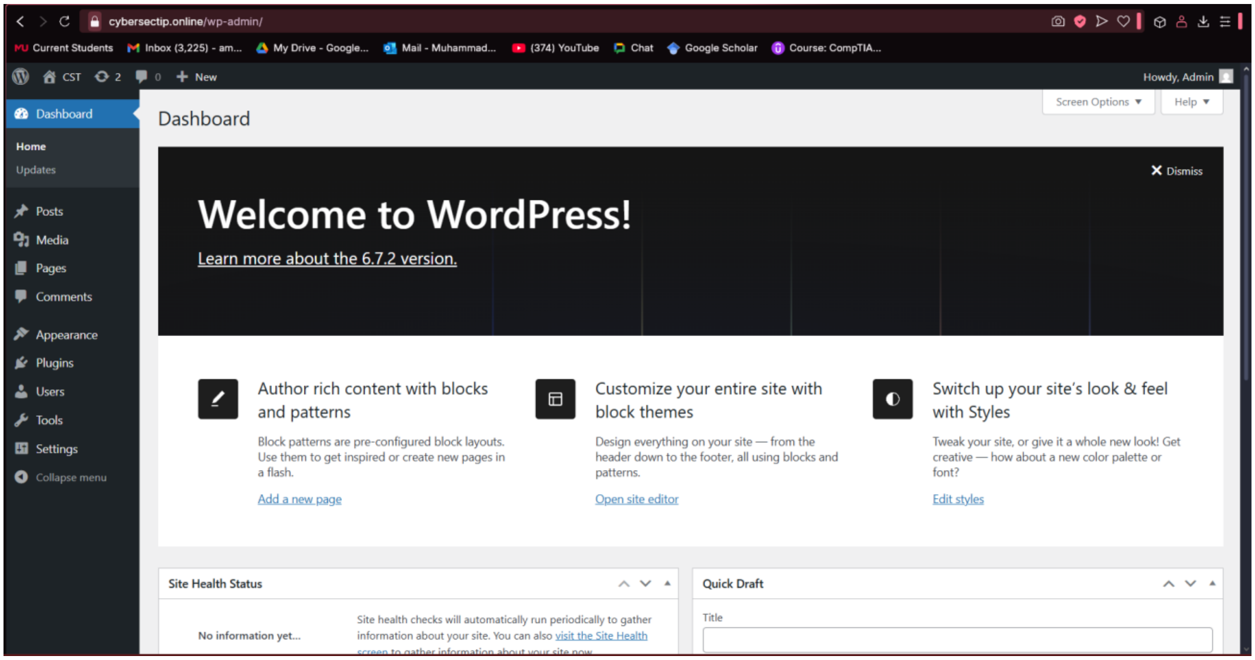
14. Now all the owners should be www-data:



```
azureuser@Server:/var/www/html$ sudo chown -R www-data:www-data *
azureuser@Server:/var/www/html$ ls -l
total 232
-rw-r--r--  1 www-data www-data   405 Feb  6  2020 index.php
-rw-r--r--  1 www-data www-data 19915 Jan  1  2024 license.txt
-rw-r--r--  1 www-data www-data  7409 Jun 18  2024 readme.html
-rw-r--r--  1 www-data www-data  7387 Feb 13  2024 wp-activate.php
drwxr-xr-x  9 www-data www-data  4096 Feb 11 16:11 wp-admin
-rw-r--r--  1 www-data www-data   351 Feb  6  2020 wp-blog-header.php
-rw-r--r--  1 www-data www-data  2323 Jun 14  2023 wp-comments-post.php
-rw-r--r--  1 www-data www-data  3336 Oct 15 15:24 wp-config-sample.php
drwxr-xr-x  4 www-data www-data  4096 Feb  4 21:01 wp-content
-rw-r--r--  1 www-data www-data  5617 Aug  2  2024 wp-cron.php
drwxr-xr-x 30 www-data www-data 12288 Feb 11 16:11 wp-includes
-rw-r--r--  1 www-data www-data  2502 Nov 26  2022 wp-links-opml.php
-rw-r--r--  1 www-data www-data  3937 Mar 11  2024 wp-load.php
-rw-r--r--  1 www-data www-data 51367 Sep 30  2024 wp-login.php
-rw-r--r--  1 www-data www-data  8543 Sep 18  2024 wp-mail.php
-rw-r--r--  1 www-data www-data 29032 Sep 30  2024 wp-settings.php
-rw-r--r--  1 www-data www-data 34385 Jun 19  2023 wp-signup.php
-rw-r--r--  1 www-data www-data  5102 Oct 18 15:56 wp-trackback.php
-rw-r--r--  1 www-data www-data  3246 Mar  2  2024 xmlrpc.php
```

15. Now access your site via your IP or DNS and be greeted by the Wordpress wizard, sleect english and continue.
16. Now to finalise this lets create a mysql database and user, first lets make our sql databse set up secure by entering: **sudo mysql_secure_installation**, say yes to all the prompts.
17. Lets enter mysql by typing: **sudo mysql**
18. To create a database lets type: **CREATE DATABASE wordpress_db;**
19. To create a user type: **CREATE USER 'wordpress_user'@'localhost' IDENTIFIED BY 'P@55word';**
20. Lets grant all privilegs by: **GRANT ALL PRIVILEGES ON wordpress_db.* TO 'wordpress_user'@'localhost';**
21. And exit: **exit;**
22. Now the wordpress express 5 minute install window should open, add your **site name, username and password** and start.

23. Once done you should see the dashboard!



Dashboard

Welcome to WordPress!

Learn more about the 6.7.2 version.

Author rich content with blocks and patterns

Block patterns are pre-configured block layouts. Use them to get inspired or create new pages in a flash.

Add a new page

Customize your entire site with block themes

Design everything on your site — from the header down to the footer, all using blocks and patterns.

Open site editor

Switch up your site's look & feel with Styles

Tweak your site, or give it a whole new look! Get creative — how about a new color palette or font?

Edit styles

Site Health Status

No information yet...

Site health checks will automatically run periodically to gather information about your site. You can also visit the Site Health screen to gather information about your site now.

Quick Draft

Title

Scripting component

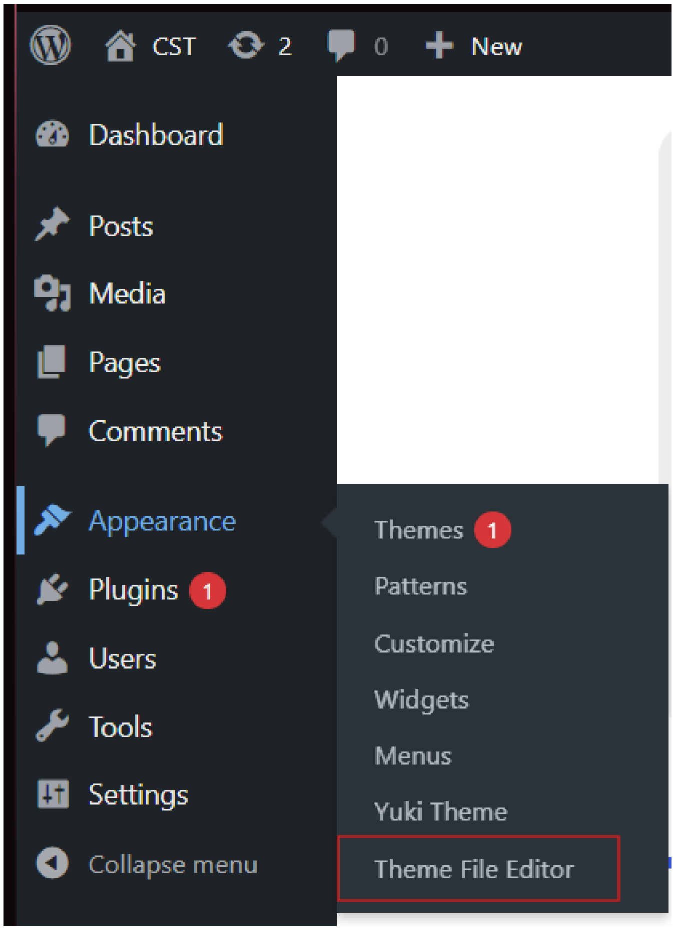
I am writing a script that helps in the security of the end user, as this is a cybersecurity tips website, it made sense to have a key component in practice.

The script will automatically load the website in HTTPS instead of HTTPS keeping it encrypted and secure.

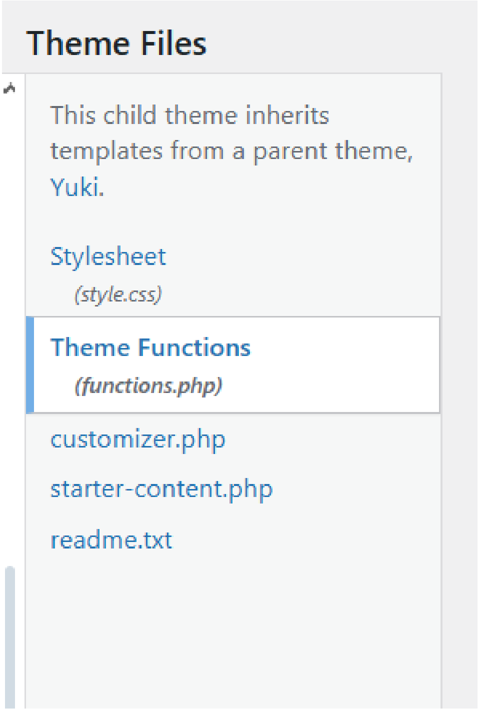
Got the idea from Stack overflow (will be in references)

Modified it to fit my needs as I am writing the Js code right inside the wordpress dashboard, here are the steps:

1. Open Wordpress dashboard
2. In appearance click on theme file editor



3. From the right hand menu click on functions.php (this is the code we will modify)



Theme Files

This child theme inherits templates from a parent theme, Yuki.

Stylesheet
(style.css)

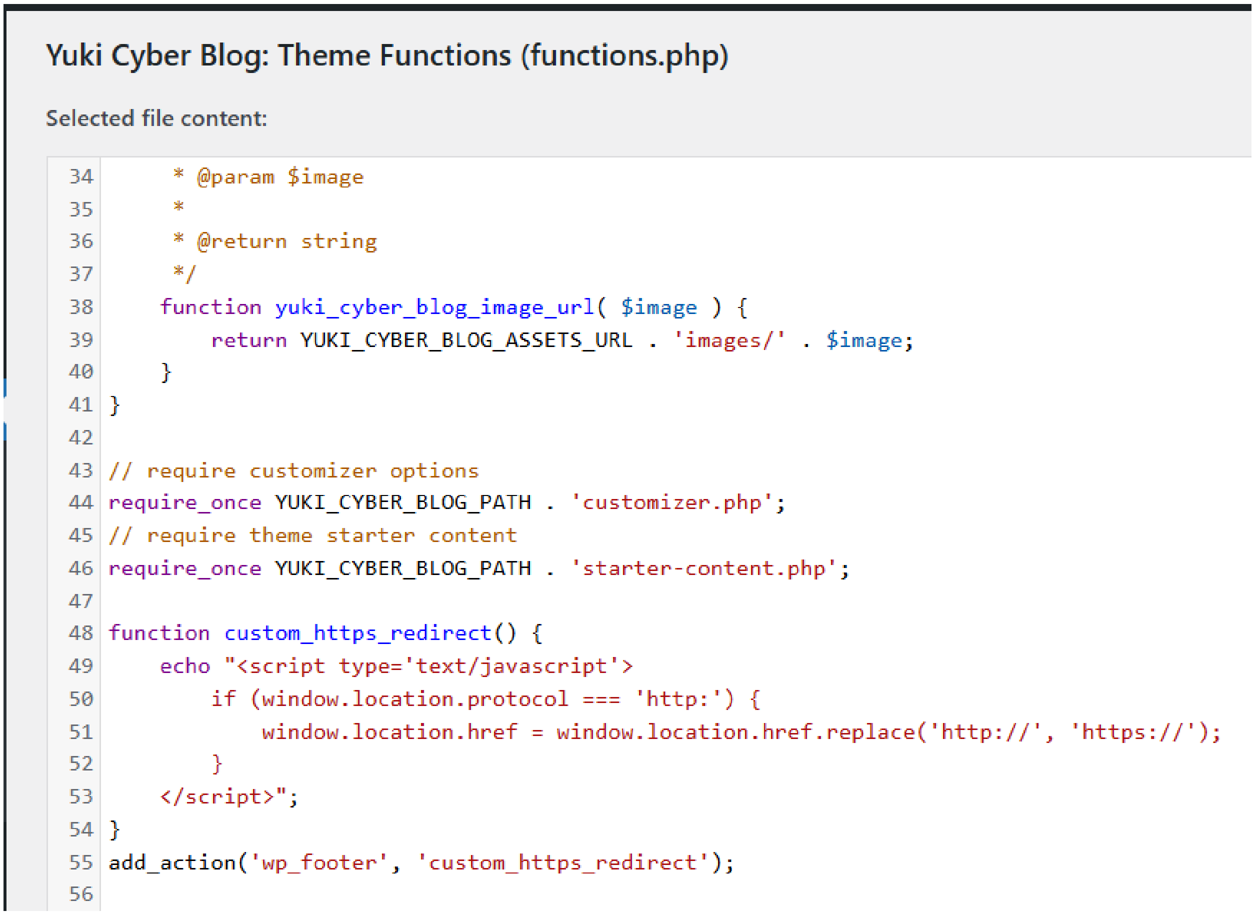
Theme Functions
(functions.php)

customizer.php

starter-content.php

readme.txt

4. At the bottom of the code add the script:



Yuki Cyber Blog: Theme Functions (functions.php)

Selected file content:

```
     * @param $image
     *
     * @return string
     */
    function yuki_cyber_blog_image_url( $image ) {
        return YUKI_CYBER_BLOG_ASSETS_URL . 'images/' . $image;
    }
}

// require customizer options
require_once YUKI_CYBER_BLOG_PATH . 'customizer.php';
// require theme starter content
require_once YUKI_CYBER_BLOG_PATH . 'starter-content.php';

function custom_https_redirect() {
    echo "<script type='text/javascript'>
        if (window.location.protocol === 'http:') {
            window.location.href = window.location.href.replace('http://', 'https://');
        }
    </script>";
}
add_action('wp_footer', 'custom_https_redirect');

```

Code explanation:

Custom_https_redirect() is a custom function that redirects http requests to https.

The code is echoed in the function which is generated at the footer, add_action then hooks the code to the footer, so when wordpress starts to render the footer on launch it will call on the function to run it.

The if statement checks if the URL equals to HTTP.

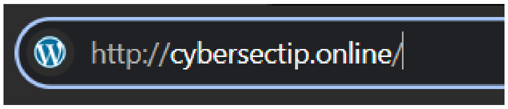
If it does it exchanges it with HTTPS.

This is how we ensure the site is always encrypted.

Verification

This code can easily be verified by testing in the search bar.

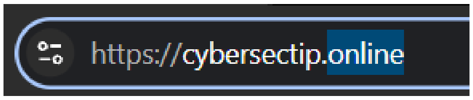
Type: http://cybersectip.online/ into your search bar like this:



http://cybersectip.online/

Press enter.

It should automatically redirect to the https site (This will be further demonstrated in the video.)



https://cybersectip.online

References

For installing Wordpress:

Abstract Programmer (2024, April 2). *How to install WordPress on Ubuntu 22.04 with Nginx*. YouTube.

https://www.youtube.com/watch?v=1Haj2D_WTCY

For the scripting:

Stack Overflow (n.d.). *Detect http or https then force https in JavaScript*. Stack Overflow.

https://stackoverflow.com/questions/4723213/detect-http-or-https-then-force-https-in-javascript