ICT 171 – ASSIGNMENT 2

WEBSITE & CLOUD SERVER DOCUMENTATION

PRINCE BEJOY

35321466

TABLE OF CONTENTS

1) INTRODUCTION:

DETAILS:

- **GLOBAL (ELASTIC) IP ADDRESS:** 18.132.89.126
- **WEBSITE DOMAIN NAME:** https://securepost.space/

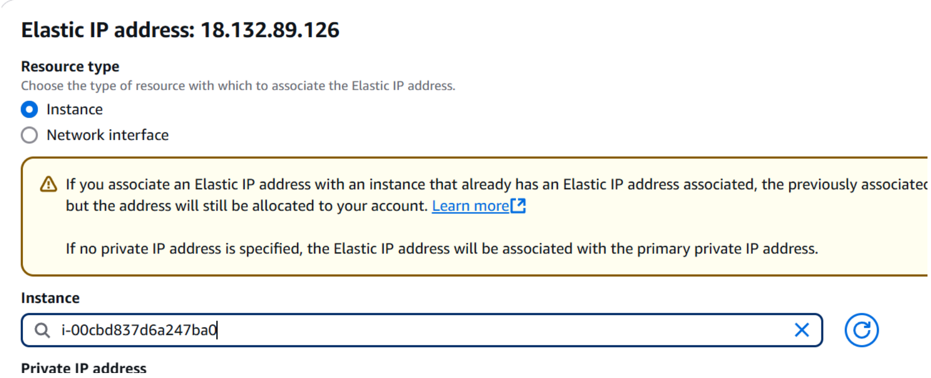
WEBSITE DESCRIPTION & WHY IAAS was used

The website that I have created is a blogging platform model named **SecurePost**, that promises greater security, stability and interactivity than competing platforms.

I have chosen **Infrastructure as a Service (IaaS)**, specifically **AWS EC2**, because it provides cost-effective scalability, automatic backups, and robust security features. Even though I use the free tier, unlike traditional hosting (i.e. WordPress), **IaaS allows complete control over the server environment**, enabling better resource management, high availability, and efficient performance, making it ideal for a dynamic and growing website like SecurePost.

2) CREATING THE INSTANCE & ESTABLISHING THE WEB SERVER (AWS EC2)

1. Ubuntu (AMI: 22.04 LTS) Instance created on EC2, type t2.micro. Associate the instance with an **elastic IP (so that the IP remains fixed)**



Elastic IP address: 18.132.89.126

Resource type
Choose the type of resource with which to associate the Elastic IP address.

● Instance
○ Network interface

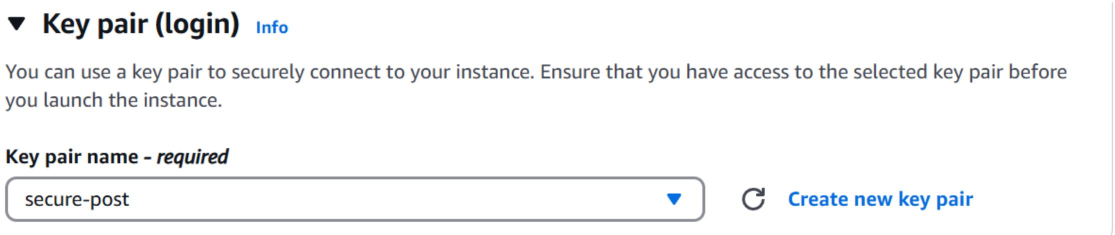
⚠ If you associate an Elastic IP address with an instance that already has an Elastic IP address associated, the previously associated but the address will still be allocated to your account. Learn more

If no private IP address is specified, the Elastic IP address will be associated with the primary private IP address.

Instance
i-00cbd837d6a247ba0

Private IP address

2. Save the key pair of the instance in the PC (.pem file)

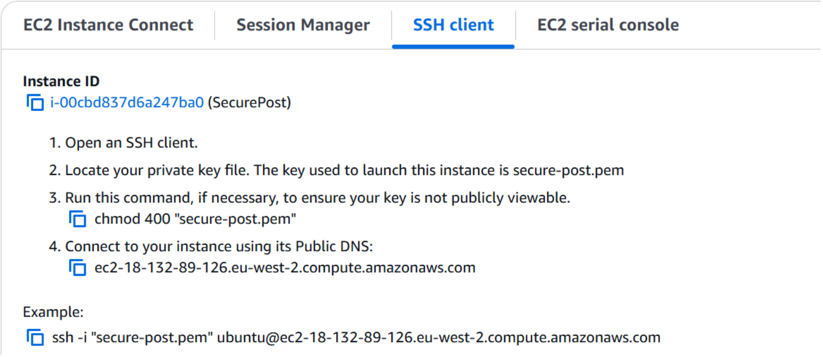


▼ Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*
secure-post

3. Connect to the SSH client using the command specified:



Instance ID
i-00cbd837d6a247ba0 (SecurePost)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is secure-post.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
```
chmod 400 "secure-post.pem"
```
4. Connect to your instance using its Public DNS:
```
ec2-18-132-89-126.eu-west-2.compute.amazonaws.com
```

Example:
```
ssh -i "secure-post.pem" ubuntu@ec2-18-132-89-126.eu-west-2.compute.amazonaws.com
```

COMMAND LINE SCRIPT (taken from https://runcloud.io/blog/aws)

4. Update system package: `sudo apt update && sudo apt upgrade -y`

Use the LAMP software stack: Apache2, mySQL (database), PHP

5. Installing Apache: `sudo apt install apache2 -y`
6. Installing mySQL: `sudo apt install mysql-server -y`
7. Installing PHP (Hypertext Processor) :

```
sudo apt install php libapache2-mod-php php-mysql php-curl
php-xml php-mbstring -y
```

8. Start and enable the installed services:

```
sudo systemctl enable apache2
sudo systemctl enable mysql
sudo systemctl restart apache2
```

9. Log into mySQL server and change plugin to *mysql_native_password*. Set up a strong password (not revealing). Create a WORDPRESS DATABASE

   `sudo mysql -u root` (root is the name of the root user)

   ➔ `ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password by 'Password';` (*password not mentioned for privacy reasons)*

   ➔ `CREATE USER ‘Wp_user'@localhost IDENTIFIED BY ‘Password’;`

   ➔ `CREATE DATABASE Wp_database;` (WordPress database)

   ➔ `GRANT ALL PRIVILEGES ON wordpress.* TO 'Wp_user'@localhost;`

10. After creating a database on WordPress, we have to install it via the SSH client. These steps are specified in the next stage

3) USING WORDPRESS

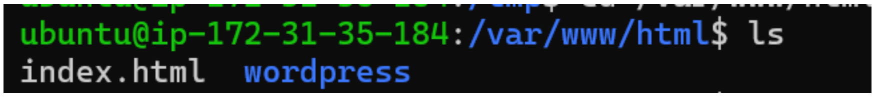
WordPress provides a user-friendly interface, making it easy to create, manage, and publish blog posts without needing advanced coding skills. Other features include:

- **Customization**
- **Security & SEO**
- **Community Support**

COMMAND LINE SCRIPT

11. Change the directory of the current script (lead to a temporary TMP directory on Linux): `cd /tmp`
12. Download the zip file: `wget https://wordpress.org/latest.tar.gz`
13. Unzip this file to access contents: `tar -xvf latest.tar.gz`
14. Shift the WordPress folder just made to the Apache server root: `sudo mv wordpress/ /var/www/html`
15. Double check if stored:

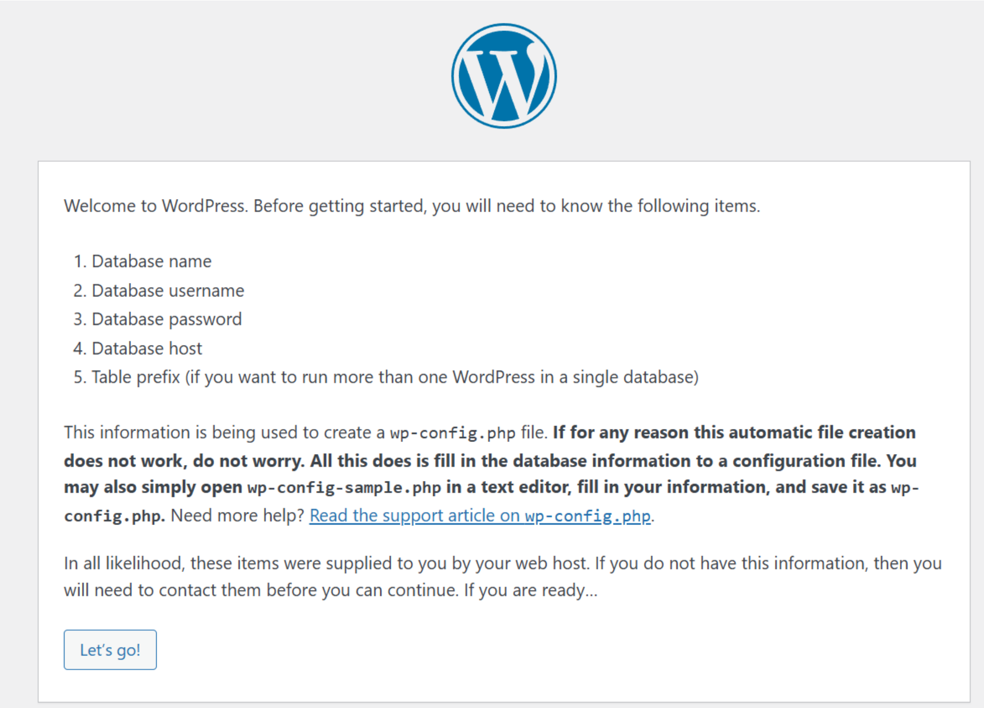
```
cd /var/www/html/
ls
```



```
ubuntu@ip-172-31-35-184:/var/www/html$ ls
index.html  wordpress
```

After the CMD script we need to modify the database details from our web browser:

16. **Enter the Public IP address followed by /wordpress. This screen is loaded:**



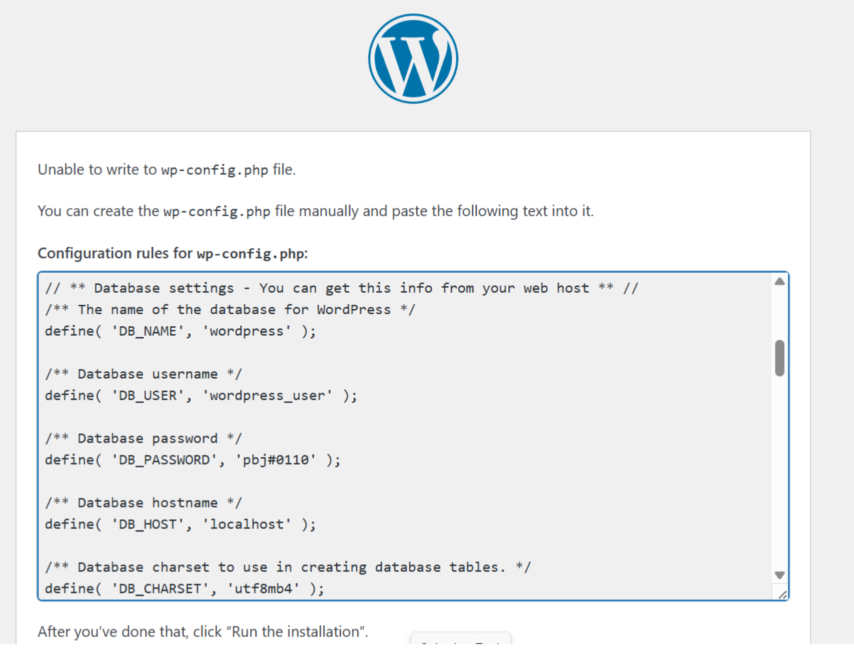
Welcome to WordPress. Before getting started, you will need to know the following items.

1. Database name
2. Database username
3. Database password
4. Database host
5. Table prefix (if you want to run more than one WordPress in a single database)

This information is being used to create a `wp-config.php` file. **If for any reason this automatic file creation does not work, do not worry. All this does is fill in the database information to a configuration file. You may also simply open `wp-config-sample.php` in a text editor, fill in your information, and save it as `wp-config.php`.** Need more help? Read the support article on `wp-config.php`.

In all likelihood, these items were supplied to you by your web host. If you do not have this information, then you will need to contact them before you can continue. If you are ready…

17. **Click "Let's go!"**
18. **Enter the details as from step (i)**
19. However, an error message will be displayed after saying that **unable to write to wp-config.php file.**



Unable to write to `wp-config.php` file.

You can create the `wp-config.php` file manually and paste the following text into it.

Configuration rules for `wp-config.php`:

```
// ** Database settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress' );

/** Database username */
define( 'DB_USER', 'wordpress_user' );

/** Database password */
define( 'DB_PASSWORD', 'pbj#0110' );

/** Database hostname */
define( 'DB_HOST', 'localhost' );

/** Database charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8mb4' );
```

After you've done that, click "Run the installation".

20. Follow the steps as specified on the page.

STEPS

1. Change the directory to the WordPress path:

```
cd /var/www/html/wordpress
```

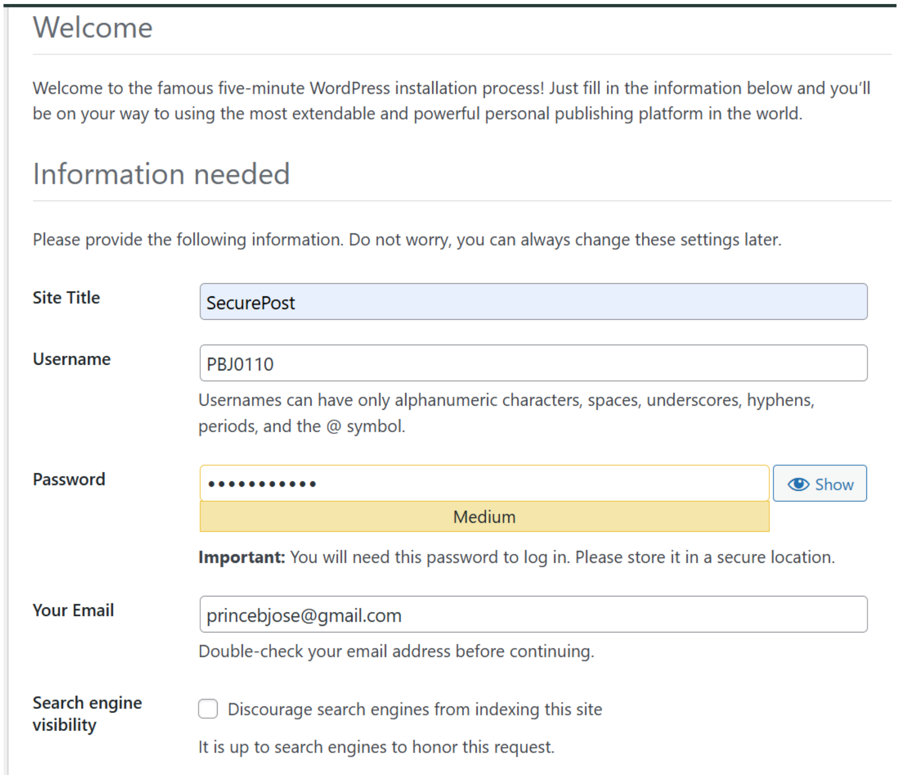
2. Edit the php file using the command: `nano wp-config.php`
3. Copy-paste the code on the page onto CMD

21. **After applying the changes to the .php, we need to provide these permissions as well:**

```
sudo chown -R www-data:www-data /var/www/html/
sudo chmod -R 755 /var/www/html/
```

22. **Restart server for changes to take effect:** `sudo systemctl restart apache2`

23. After this, refresh the path on the address bar, A screen will be displayed:



Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Do not worry, you can always change these settings later.

Site Title: SecurePost

Username: PBJ0110

Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password: ••••••••••• Show

Medium

Important: You will need this password to log in. Please store it in a secure location.

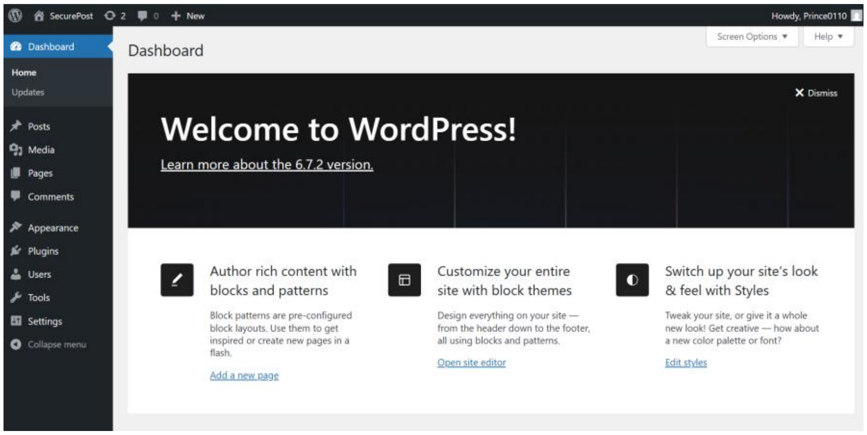
Your Email: princebjose@gmail.com

Double-check your email address before continuing.

Search engine visibility: □ Discourage search engines from indexing this site

It is up to search engines to honor this request.

24. Enter pertaining details and login to WordPress successfully.



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.

Customize your entire site with block themes

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

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?

25. ADMIN ACCOUNT SUCCESSFULLY CREATED!
26. We need to change the Apache config so that sub path of the website is only the IP address.
27. This can be done by:

    Check available configurations: `cd /etc/apache2/sites-available/`

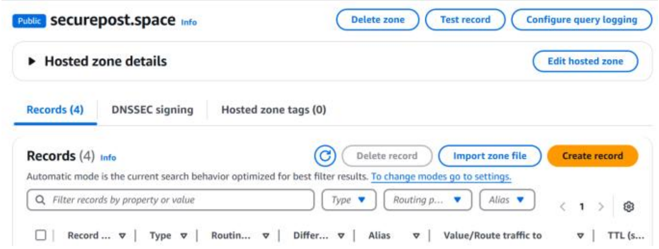
    ```
    ls
    sudo nano 000-default.conf
    ```

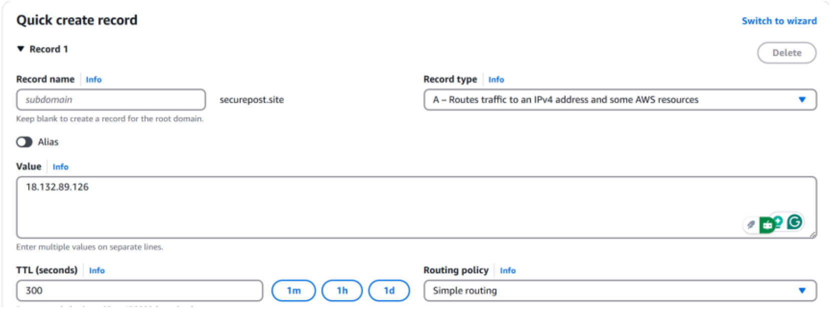
    Replace the DocumentRoot and add /wordpress

    Restart the apache server: `sudo systemctl restart apache2`

4) CREATING & LINKING DNS ENTRIES

28. For my project I have used GoDaddy (https://godaddy.com/) for purchasing my domain name.
29. AWS Route 53 can also be used for **creating a hosted zone, from which I can create 2 records that point to my IP address. Set a TTL of 300 seconds**





Record name: subdomain securepost.site

Keep blank to create a record for the root domain.

Alias

Record type: A – Routes traffic to an IPv4 address and some AWS resources

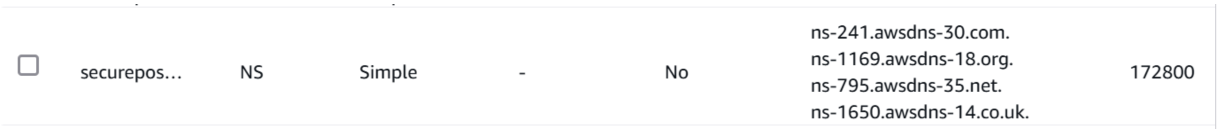
Value: 18.132.89.126

Enter multiple values on separate lines.

TTL (seconds): 300

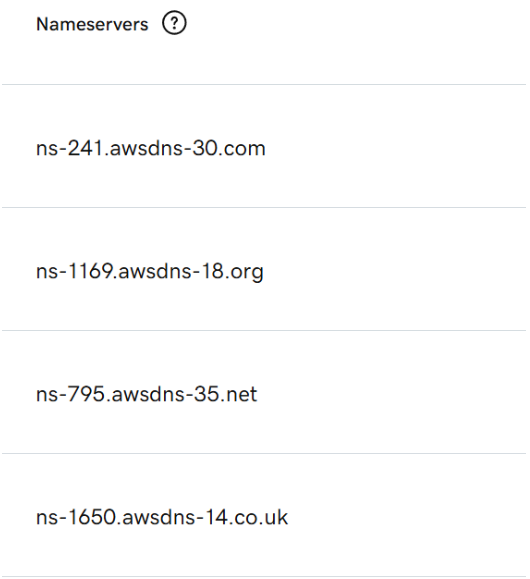
Routing policy: Simple routing

30. Records are successfully created.
31. After this, copy the route traffic from record type NS to the GoDaddy records



☐	securepos...	NS	Simple	-	No	ns-241.awsdns-30.com. ns-1169.awsdns-18.org. ns-795.awsdns-35.net. ns-1650.awsdns-14.co.uk.	172800

32. Go to **Nameservers** under the DNS records
33. Modify the existing ones to add the 4 new ones from Route53



Nameservers

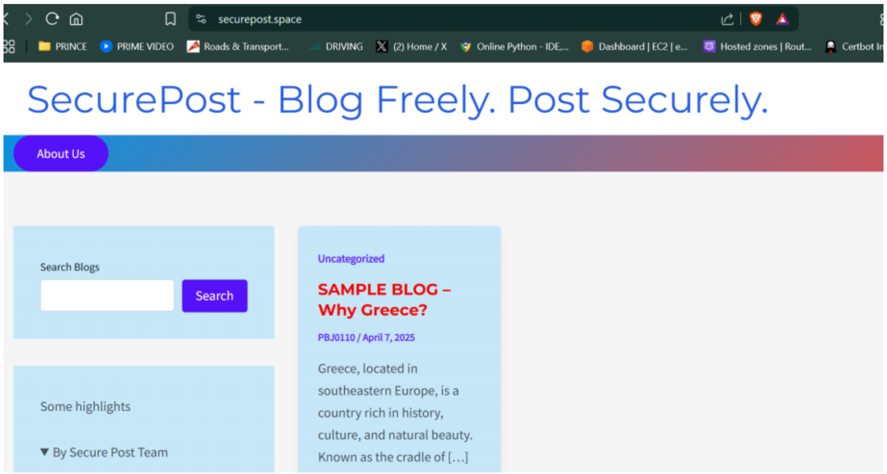
ns-241.awsdns-30.com

ns-1169.awsdns-18.org

ns-795.awsdns-35.net

ns-1650.awsdns-14.co.uk

34. After carrying out these steps, when entering www.securepost.space, the WordPress website should be displayed.



SecurePost - Blog Freely. Post Securely.

Search Blogs

Some highlights

▼ By Secure Post Team

Uncategorized

SAMPLE BLOG – Why Greece?

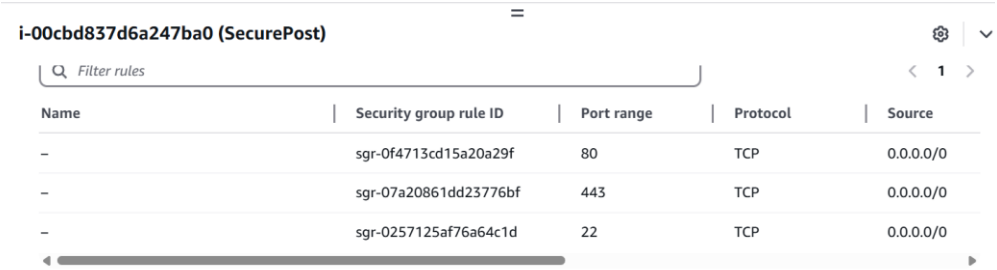
PBJ0110 / April 7, 2025

Greece, located in southeastern Europe, is a country rich in history, culture, and natural beauty. Known as the cradle of […]

5) SSL/ TLS certification

After successfully creating the website, we should be able to change the HTTP protocol to HTTPS, so that it becomes recognized as a secure website.

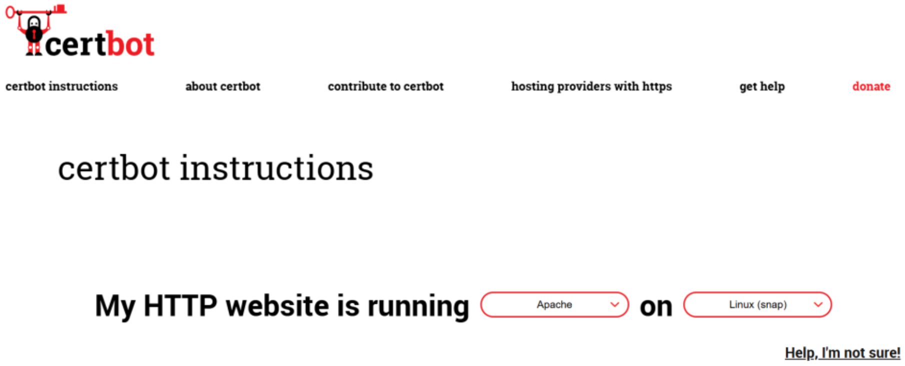
PREREQUISITES: HTTPS PORT 80, CREATED WEBSITE



i-00cbd837d6a247ba0 (SecurePost)

Name	Security group rule ID	Port range	Protocol	Source
–	sgr-0f4713cd15a20a29f	80	TCP	0.0.0.0/0
–	sgr-07a20861dd23776bf	443	TCP	0.0.0.0/0
–	sgr-0257125af76a64c1d	22	TCP	0.0.0.0/0

35. **Certbot** has to be enabled to give an HTTPS certificate to my website

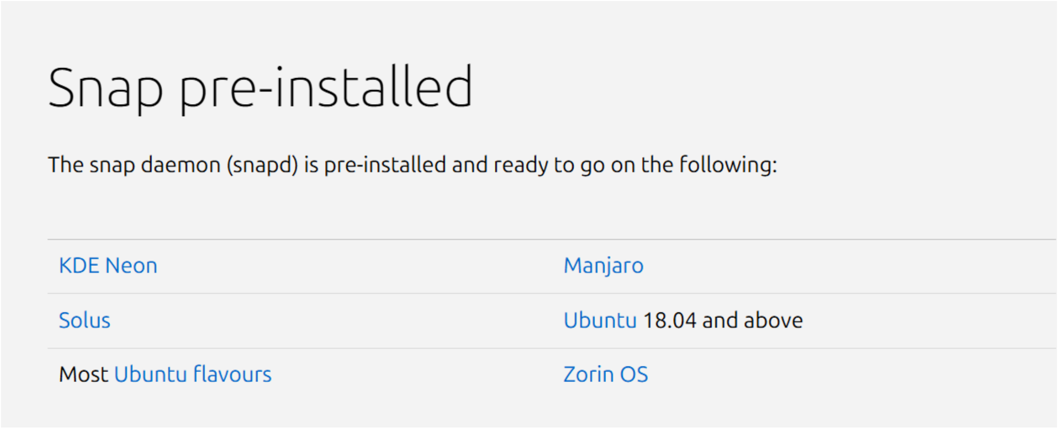


certbot instructions

My HTTP website is running Apache on Linux (snap)

36. Follow instructions on: https://certbot.eff.org/instructions?ws=apache&os=snap

37. Will ask to download snap



Snap pre-installed

The snap daemon (snapd) is pre-installed and ready to go on the following:

KDE Neon	Manjaro
Solus	Ubuntu 18.04 and above
Most Ubuntu flavours	Zorin OS

38 . If this does not work, install snap through CMD:

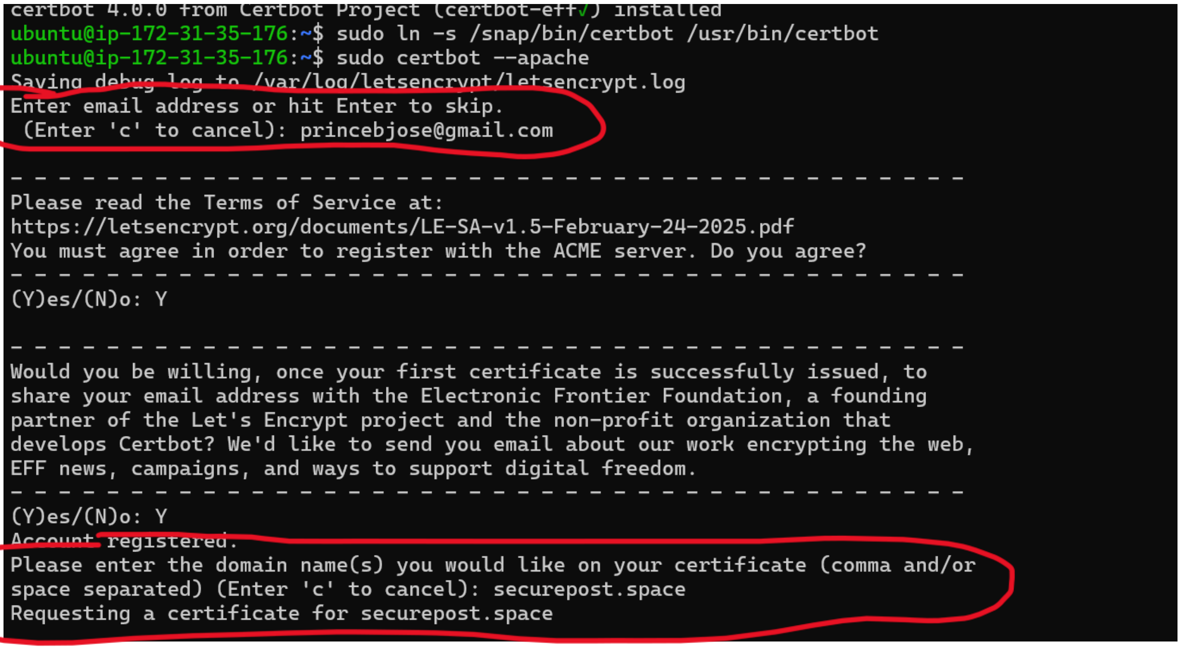
```
sudo apt install snapd -y
sudo snap install core && sudo snap refresh core
```

39. Install certbot certificate: `sudo snap install --classic certbot`

40. Create a directory link to ensure these commands can be run: `sudo ln -s /snap/bin/certbot /usr/bin/certbot`

41. Run it on the Apache server : `sudo certbot -apache`

42. Enter domain name and personal email address when prompted:



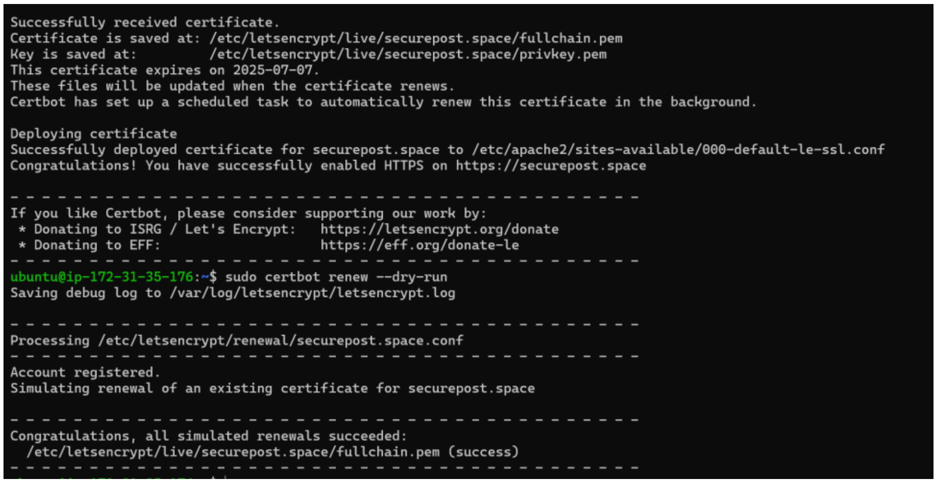
```
certbot 4.0.0 from Certbot Project (certbot-eff✓) installed
ubuntu@ip-172-31-35-176:~$ sudo ln -s /snap/bin/certbot /usr/bin/certbot
ubuntu@ip-172-31-35-176:~$ sudo certbot --apache
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Enter email address or hit Enter to skip.
 (Enter 'c' to cancel): princebjose@gmail.com

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Please read the Terms of Service at:
https://letsencrypt.org/documents/LE-SA-v1.5-February-24-2025.pdf
You must agree in order to register with the ACME server. Do you agree?
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(Y)es/(N)o: Y

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Would you be willing, once your first certificate is successfully issued, to
share your email address with the Electronic Frontier Foundation, a founding
partner of the Let's Encrypt project and the non-profit organization that
develops Certbot? We'd like to send you email about our work encrypting the web,
EFF news, campaigns, and ways to support digital freedom.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(Y)es/(N)o: Y
Account registered.
Please enter the domain name(s) you would like on your certificate (comma and/or
space separated) (Enter 'c' to cancel): securepost.space
Requesting a certificate for securepost.space
```

43. Make sure the certificate can be automatically renewed: `sudo certbot renew --dry-run`

44. **CERTIFICATE IS SUCCESSFULLY CREATED**



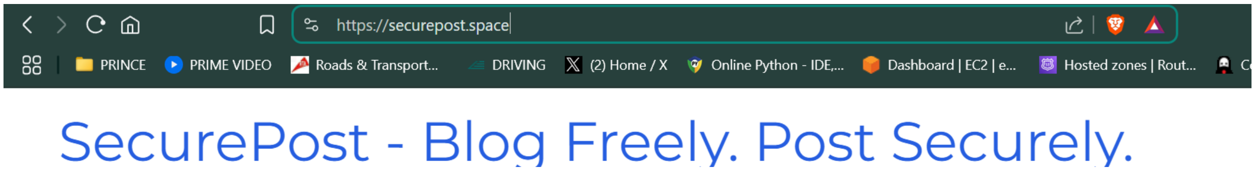
```
Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/securepost.space/fullchain.pem
Key is saved at:         /etc/letsencrypt/live/securepost.space/privkey.pem
This certificate expires on 2025-07-07.
These files will be updated when the certificate renews.
Certbot has set up a scheduled task to automatically renew this certificate in the background.

Deploying certificate
Successfully deployed certificate for securepost.space to /etc/apache2/sites-available/000-default-le-ssl.conf
Congratulations! You have successfully enabled HTTPS on https://securepost.space

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
If you like Certbot, please consider supporting our work by:
 * Donating to ISRG / Let's Encrypt:   https://letsencrypt.org/donate
 * Donating to EFF:                    https://eff.org/donate-le
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
ubuntu@ip-172-31-35-176:~$ sudo certbot renew --dry-run
Saving debug log to /var/log/letsencrypt/letsencrypt.log

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Processing /etc/letsencrypt/renewal/securepost.space.conf
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Account registered.
Simulating renewal of an existing certificate for securepost.space

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Congratulations, all simulated renewals succeeded:
  /etc/letsencrypt/live/securepost.space/fullchain.pem (success)
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```



SecurePost - Blog Freely. Post Securely.

6) References & Conclusion

Conclusion

In this project, I successfully deployed **SecurePost**, a secure and customizable blogging platform built on WordPress, hosted on an AWS EC2 instance. The website features a MySQL database, an Apache server, and WordPress integration, allowing real-time content updates via the admin panel. I implemented security best practices, ensured proper file permissions, configured the .htaccess and wp-config.php files, and resolved issues related to subpath image rendering. To streamline customization, I integrated GitHub for version control and theme development, and documented the entire process—including setup steps, code snippets, and screenshots—in a public GitHub repository. A video explainer walkthrough has also been included for better understanding and transparency. This assignment enhanced my skills in cloud deployment, Linux server configuration, WordPress management, and documentation—critical skills for my career in cybersecurity and IT.

References

https://www.youtube.com/watch?v=hRSj2n-XKGM: Creating Nameservers.

https://certbot.eff.org/instructions?ws=apache&os=snap: Using certbot.

https://www.youtube.com/watch?v=72BlLFdzzNM&pp=0gcJCdgA07VqN5tD: Linking IP address with my domain name.

https://books.google.com/books?hl=en&lr=&id=Ue62DwAAQBAJ&oi=fnd&pg=PR3&dq=creating+an+aws+ec2+instance&ots=qfOMswDMqu&sig=M1223gJn0XG9jaRkQg8OKk0bqz4
Creating EC2 instance

https://www.youtube.com/watch?v=184SdvzLwzs&pp=ygUkaG93IHRvIGNyZWF0ZSBhIHdlYnNpdGUgb24gd29yZHByZXNz Modifying website on WordPress.