

Deploying WordPress on Ubuntu-23.10 and Implementing Website Log Monitoring

Report Prepared For



Report issued: 19/09/2024

Submitted by: ABHINAV.S

Sensitive: The information in this document is strictly confidential and is made by ABHINAV.S

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 ubuntu-23.10 and setting up easy log monitoring with Splunk:

➤ Step1 : installing ubuntu server on virtualBox

For the initial step, install the Ubuntu Server either directly from Chrome or use the provided link below.

Once the Ubuntu Server is installed, proceed to set it up within VirtualBox for the subsequent steps.

➤ Step2: Update System Packages

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

```
ubuntu@ubuntu: ~$ sudo -i
```

```
root@ubuntu: ~# apt update
```

```
root@ubuntu: ~# apt upgrade
```

➤ Step3: install Apache

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

```
root@ubuntu: ~# apt install apache2
```

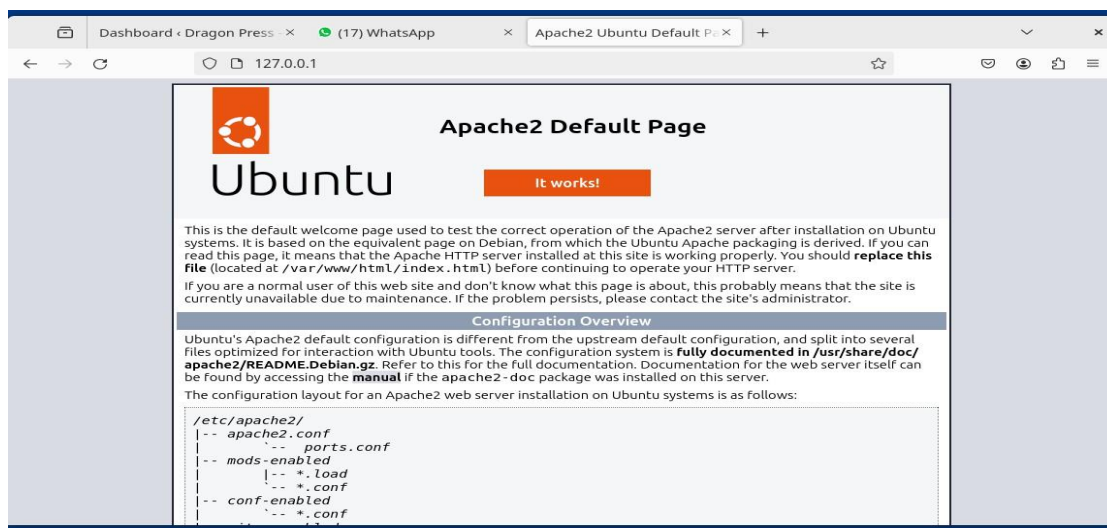
To confirm that Apache is installed on your system, execute the following command.

root@ubuntu: ~# systemctl status apache2

```
root@ubuntu: ~  
● apache2.service - The Apache HTTP Server  
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enabled)  
   Active: active (running) since Thu 2024-04-04 09:44:28 UTC; 5min ago  
     Docs: https://httpd.apache.org/docs/2.4/  
   Process: 1221 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)  
   Main PID: 1288 (apache2)  
     Tasks: 6 (limit: 7830)  
  Memory: 21.8M  
    CPU: 470ms  
   CGroup: /system.slice/apache2.service  
           └─1288 /usr/sbin/apache2 -k start  
             └─1292 /usr/sbin/apache2 -k start  
               └─1293 /usr/sbin/apache2 -k start  
                 └─1294 /usr/sbin/apache2 -k start  
                   └─1295 /usr/sbin/apache2 -k start  
                     └─1297 /usr/sbin/apache2 -k start  
  
Apr 04 09:44:27 ubuntu systemd[1]: Starting apache2.service - The Apache HTTP Server...  
Apr 04 09:44:28 ubuntu apachectl[1270]: AH00558: apache2: Could not reliably determine the server's fu  
Apr 04 09:44:28 ubuntu systemd[1]: Started apache2.service - The Apache HTTP Server.  
...  
...  
[lines 1-20/20 (END)]
```

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

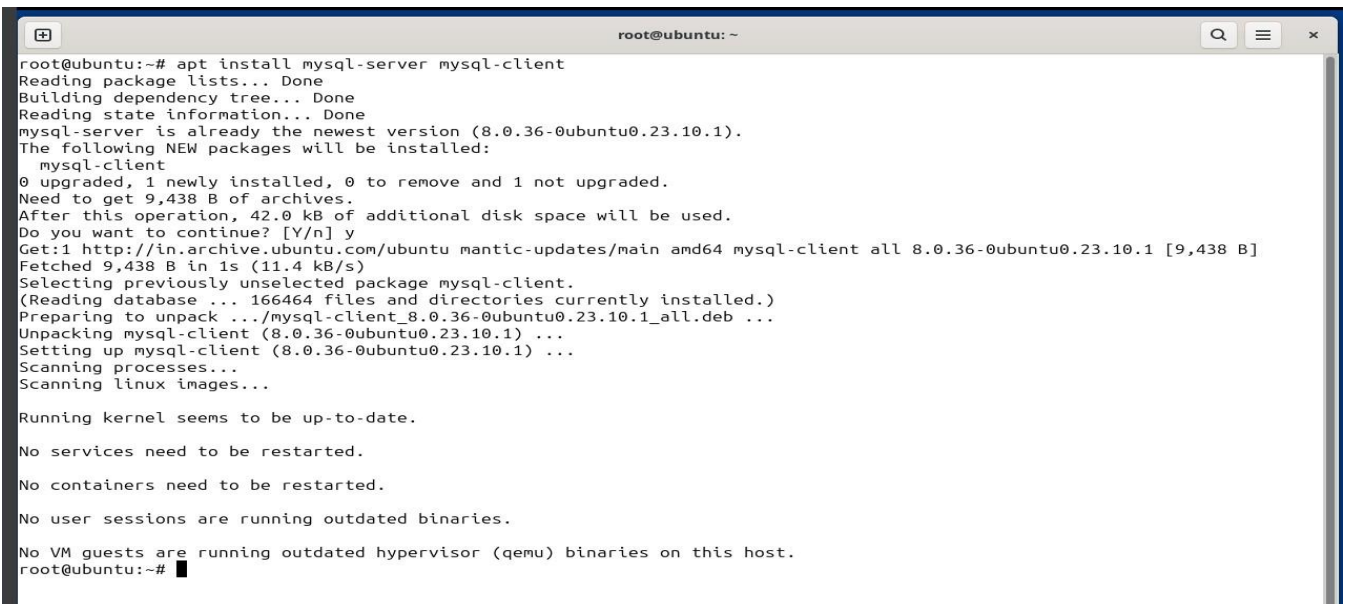
<https://ip-address>



➤ Step 4 :MySQL Server

Install the MySQL server on your Ubuntu system:

```
root@ubuntu: ~# apt install mysql-server mysql-client
```

A terminal window titled 'root@ubuntu: ~' showing the output of the command 'apt install mysql-server mysql-client'. The output indicates that mysql-server is already the newest version (8.0.36-0ubuntu0.23.10.1) and that the following NEW packages will be installed: mysql-client. It shows the download of mysql-client from the Ubuntu archive, unpacking, and setting up. The terminal also shows that no services need to be restarted and no user sessions are running outdated binaries.

```
root@ubuntu:~# apt install mysql-server mysql-client
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
mysql-server is already the newest version (8.0.36-0ubuntu0.23.10.1).
The following NEW packages will be installed:
  mysql-client
0 upgraded, 1 newly installed, 0 to remove and 1 not upgraded.
Need to get 9,438 B of archives.
After this operation, 42.0 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu mantic-updates/main amd64 mysql-client all 8.0.36-0ubuntu0.23.10.1 [9,438 B]
Fetched 9,438 B in 1s (11.4 kB/s)
Selecting previously unselected package mysql-client.
(Reading database ... 166464 files and directories currently installed.)
Preparing to unpack .../mysql-client_8.0.36-0ubuntu0.23.10.1_all.deb ...
Unpacking mysql-client (8.0.36-0ubuntu0.23.10.1) ...
Setting up mysql-client (8.0.36-0ubuntu0.23.10.1) ...
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.
root@ubuntu:~#
```

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:

```
root@ubuntu: ~# mysql_secure_installation
```

➤ Step 5 : Install PHP

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

```
root@ubuntu: ~# apt install php php-mysql
```

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

```
root@ubuntu: ~# 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>

➤ 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:

```
root@ubuntu: ~# 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 'wordpressuser' 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;
```

➤ Step 7 : Download and Extract WordPress

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

```
root@ubuntu: ~# cd /tmp wget https://wordpress.org/latest.tar.gz
```

```
root@ubuntu: ~# tar xf latest.tar.gz
```

```
root@ubuntu: ~# mv wordpress /var/www/html/
```

➤ Step 8 : Set Permissions

Adjust ownership and permissions for the WordPress files:

```
root@ubuntu: ~# chown -R www-data: www-data /var/www/html/wordpress
```

```
root@ubuntu: ~# chmod -R 755 /var/www/html/wordpress
```

➤ Step 9: Configure Apache

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

```
root@ubuntu: ~# nano /etc/apache2/sites-available/wordpress.conf
```

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

apache

Copy code

```
<VirtualHost *:80>
```

```
ServerAdmin webmaster@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}/error.log
```

```
CustomLog ${APACHE_LOG_DIR}/access.log combined
```

```
</VirtualHost>
```

Enable the virtual host Apache:

```
root@ubuntu: ~# a2ensite wordpress
```

➤ Step 10: Set Up WordPress Configuration

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

```
root@ubuntu: ~# cd /var/www/html/wordpress
```

```
root@ubuntu: ~# 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:

```
root@ubuntu: ~# 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.

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.

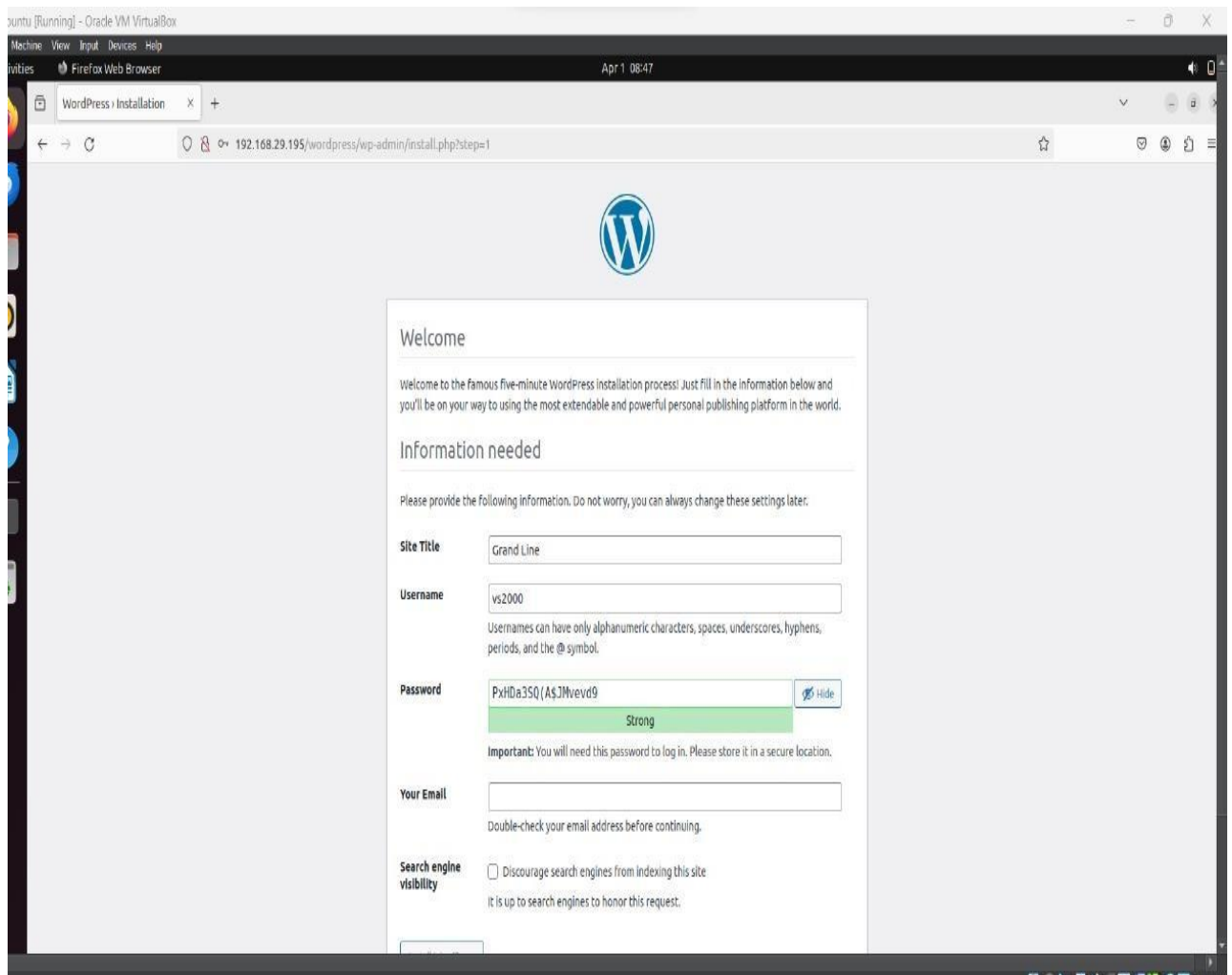
Database Host: Leave this field blank. WordPress will use the default value ('localhost'), which is appropriate for most setups.

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.

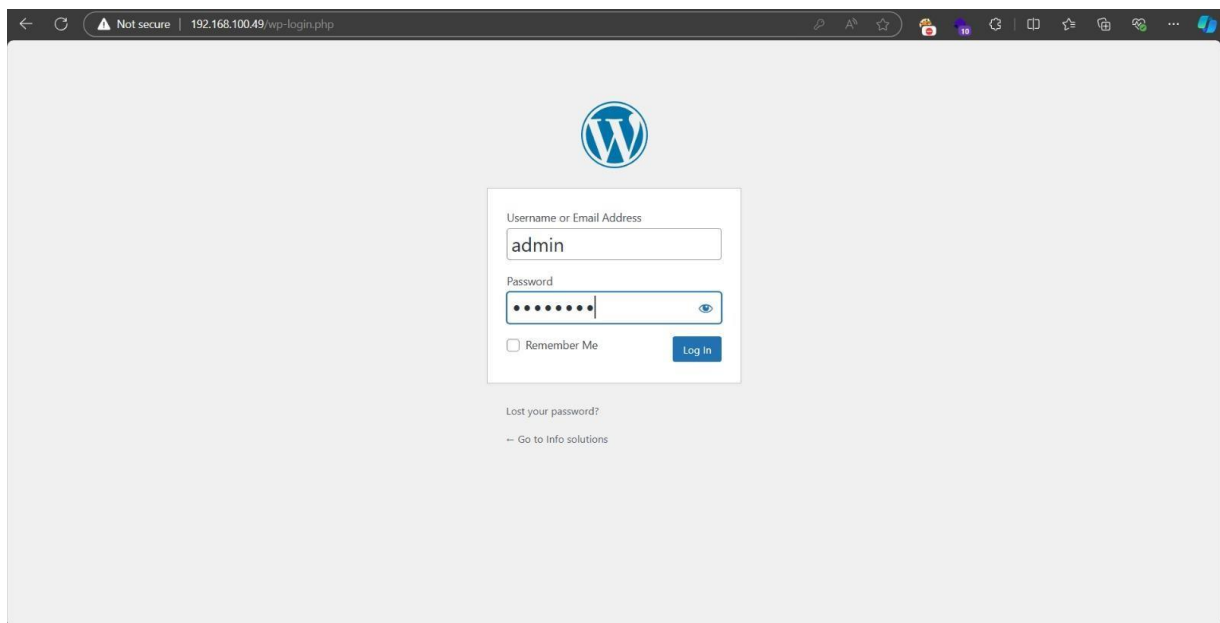
Password: Choose a strong password for the admin account.

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.

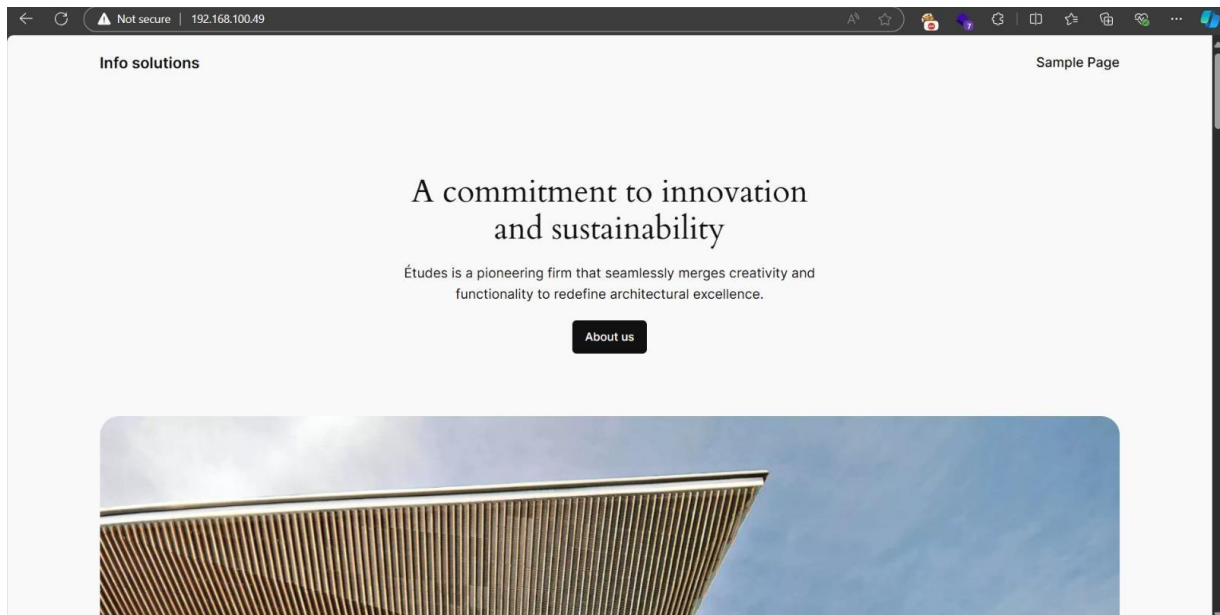


Success:

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

```
root@ubuntu: ~# 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.

```
root@ubuntu: ~# systemctl restart apache2
```

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

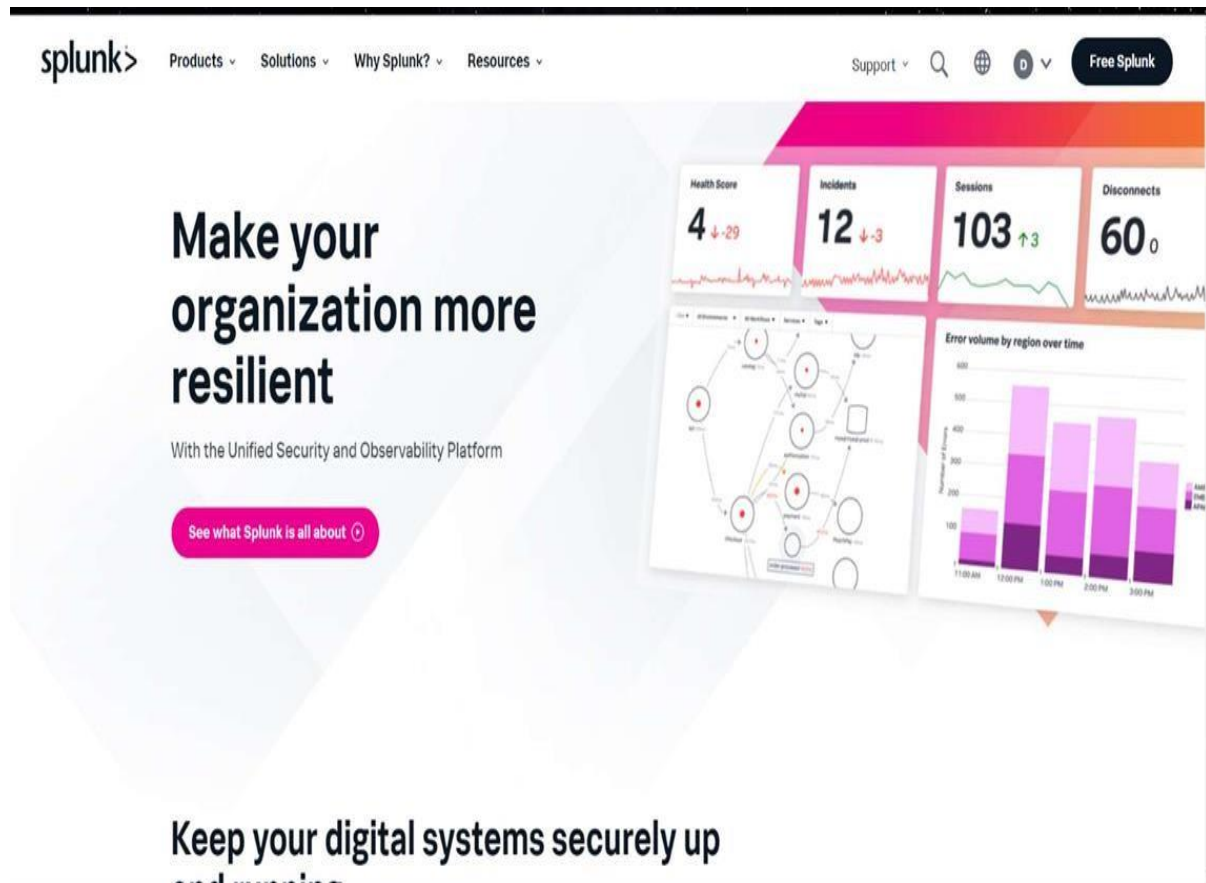
➤ Step12: Installing and Configuring Splunk Universal

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

splunk > Products ▾ Solutions ▾ Why Splunk? ▾ Resources ▾ Support ▾ 🔍 🌐 ⓘ ▾

Universal Forwarders provide reliable, secure data collection from remote sources and forward that data into Splunk software for indexing and consolidation. They can scale to tens of thousands of remote systems, collecting terabytes of data.

Choose Your Installation Package

Windows Linux Mac OS Free BSD Solaris AIX

ARM

4.14+, 5.4+ kernel Linux distributions with
libc v2.21+, Graviton+ Servers 64-bit

Package	Size	Download Now
.rpm	29.97 MB	Download Now
.deb	20.36 MB	Download Now
.tgz	29.99 MB	Download Now

PPC64

3.x+, 4.x+, or 5.x+ kernel Linux distributions

Package	Size	Download Now
.tgz	29.24 MB	Download Now
.rpm	29.28 MB	Download Now

Waiting for n2moustflow.com...

- Alternative: Download via Command Line:

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

splunk > Products ▾ Solutions ▾ Why Splunk? ▾ Resources ▾ Support ▾ 🔍 🌐 ⓘ ▾ **Free Splunk**

GET STARTED

You're Downloading Splunk Universal Forwarder 9.1.3 for Linux

Your download should have started. No? [Try this URL](#).

[Choose additional platforms here](#).

USEFUL TOOLS

- Download via Command Line
- Download via Browser

We've got ampersands in the URL and they're all escaped and ready for wget. This URL won't work in your browser. Click here to select the entire command.

```
wget -O splunkforwarder-9.1.3-d95b3299fa65-linux-armv8.deb "https://download.splunk.com/products/universalforwarder/releases/9.1.3/linux/splunkforwarder-9.1.3-d95b3299fa65-linux-armv8.deb"
```

Install + Setup

- > Universal Forwarder Manual

Use + Extend

- > Splunk Universal Forwarder Blog

Aid + Assistance

- > Splunk Answers
- > Ask an Expert

Community

- > Find a User Group
- > Splunk Developers

Download via Command Line:

Open a terminal on your Ubuntu machine.

Change to the /tmp directory:

```
root@ubuntu: ~# cd /tmp
```

Paste and run the wget command you copied:

```
root@ubuntu: ~# 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:

```
root@ubuntu: ~# tar xvfz splunkforwarder--Linux--bit.tgz -C /opt
```

Replace and with the actual version and system architecture.

```
root@ubuntu: ~# cd /opt
```

```
root@ubuntu: ~# tar xvfz /tmp/splunkforwarder--Linux--bit.tgz
```

Start Splunk Universal Forwarder

Navigate to bin Directory:

Change to the bin directory within the Splunk Forwarder installation:

```
root@ubuntu: ~# cd /opt/splunkforwarder/bin
```

Start Splunk Forwarder:

Initiate the Splunk Forwarder and accept the license agreement:

```
root@ubuntu: ~#. /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:

```
root@ubuntu:~# ./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:

```
root@ubuntu: ~# nano /opt/splunkforwarder/etc/system/local/outputs.conf
```

If the file doesn't exist, create it:

```
root@ubuntu: ~# 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 <splunk_server_ip> with your Splunk server's IP address and <splunk_listener_port>

with the designated port (e.g., 9997).

Define Monitored Logs in inputs.conf

Edit inputs.conf:

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

```
root@ubuntu: ~# nano /opt/splunkforwarder/etc/system/local/inputs.conf
```

If the file is not present, create it:

```
root@ubuntu: ~# 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 = web_logs
```

```
[monitor:///var/log/apache2/error.log]
```

```
sourcetype = apache_error
```

```
index = web_logs
```

```
[monitor:///var/log/apache2/other_vhosts_access.log]
```

```
sourcetype = apache_error
```

```
index = web_logs
```

Adjust paths and configurations according to your specific log requirements.

Restart Splunk Universal Forwarder Save Changes and Restart Splunk UF:

```
root@ubuntu: ~# /opt/splunkforwarder/bin/splunk restart
```

➤ Step 14: Adjust Firewall Settings for Splunk UF Communication

Check Firewall Status:

```
root@ubuntu: ~# ufw status
```

If the firewall is inactive, enable it:

```
root@ubuntu: ~# ufw enable
```

Allow Splunk UF Traffic:

```
root@ubuntu: ~# ufw allow <splunk_listener_port>/tcp
```

Replace with the specific port (e.g., 9997) designated for sending logs.

Optional: Allow Additional Ports (if required):

```
root@ubuntu: ~# ufw allow 80/tcp
```



```
root@ubuntu: ~# ufw allow 22
```

```
root@ubuntu: ~# ufw allow 443/tcp
```

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

```
root@ubuntu: ~# ufw reload
```

Restart Splunk Universal Forwarder:

```
root@ubuntu: ~# /opt/splunkforwarder/bin/splunk restart
```

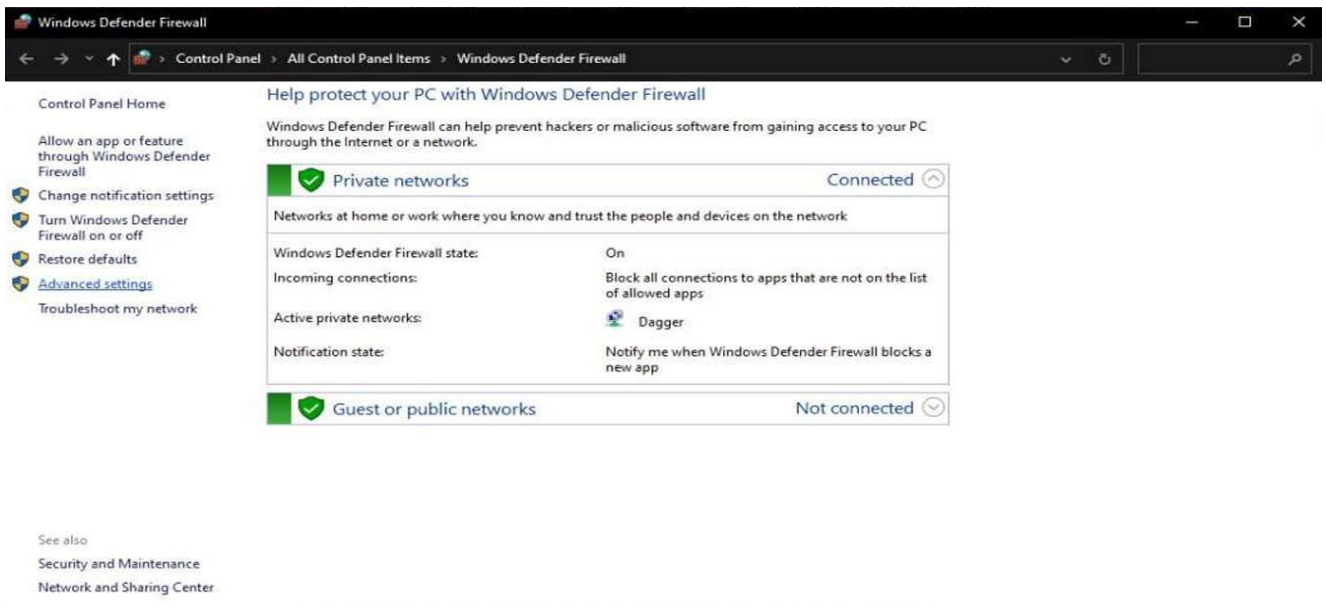
These steps provide detailed instructions for configuring the Splunk Universal Forwarder on an Ubuntu server, ensuring proper log monitoring and firewall settings for effective communication with the Splunk server.

➤ STEP 15: Preparing Splunk Server and Connecting it to Splunk Enterprise for WordPress Log Tracking

Configuring Windows Firewall for Splunk Universal Forwarder:

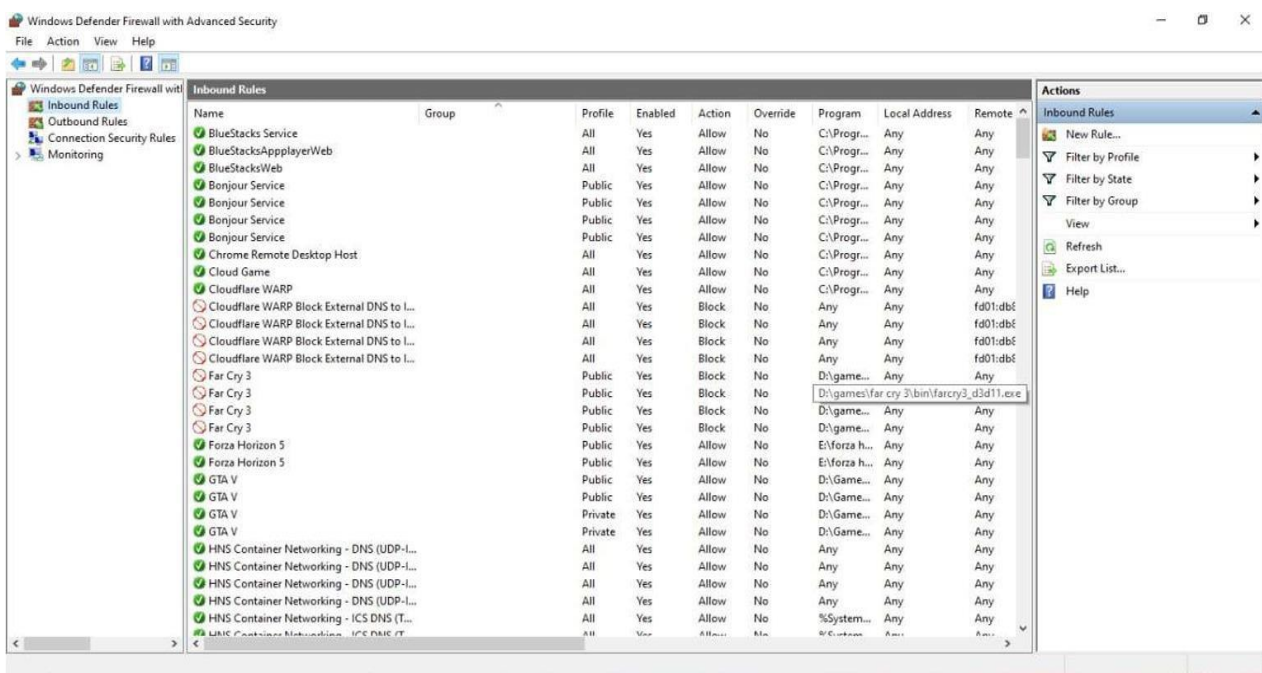
- Access Windows Firewall Settings:

Navigate to Windows settings and select Windows Defender Firewall.



- Navigate to Advanced Settings:

Click on "Advanced settings" in the left panel.

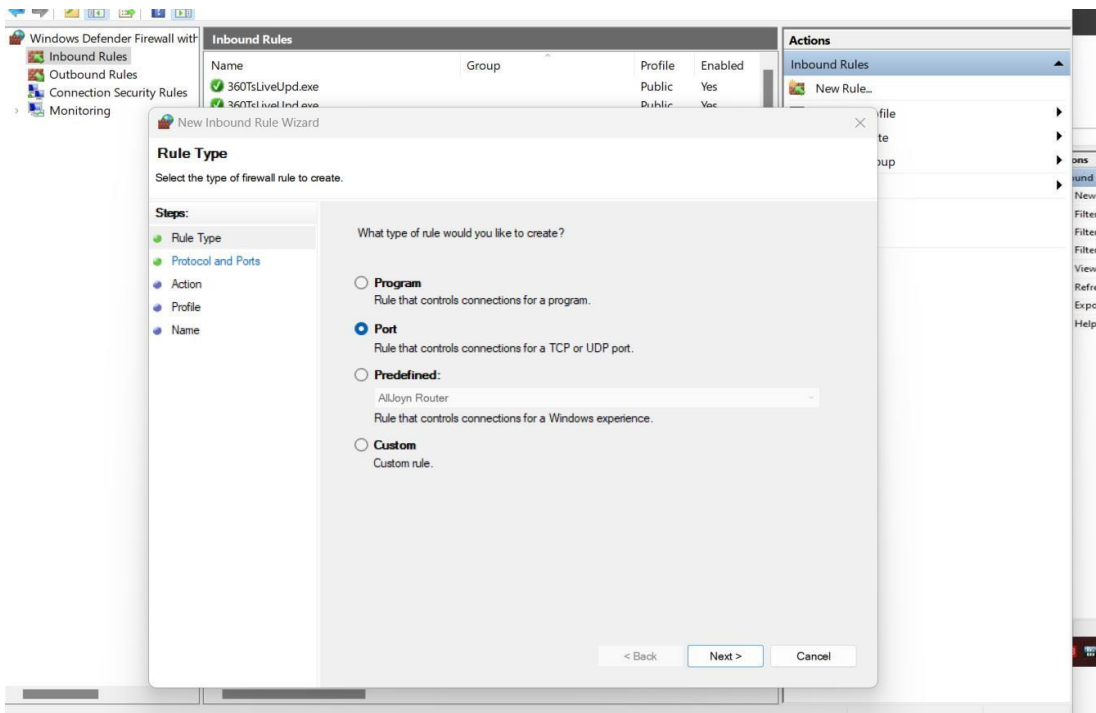


- Create Inbound Rule:

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."

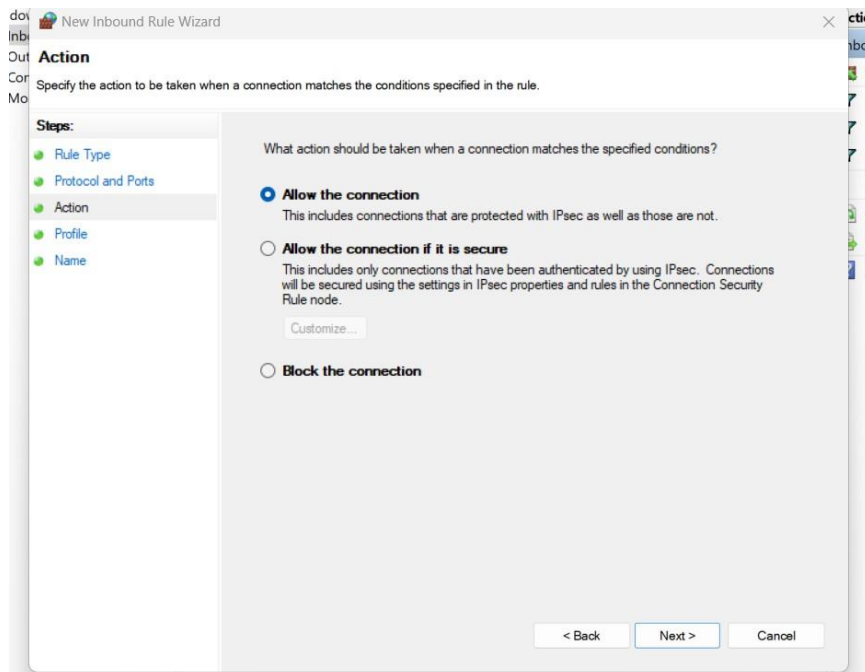


- Specify Port and Protocol:

Choose "TCP" and enter the assigned before (e.g., 9997), then click "Next."

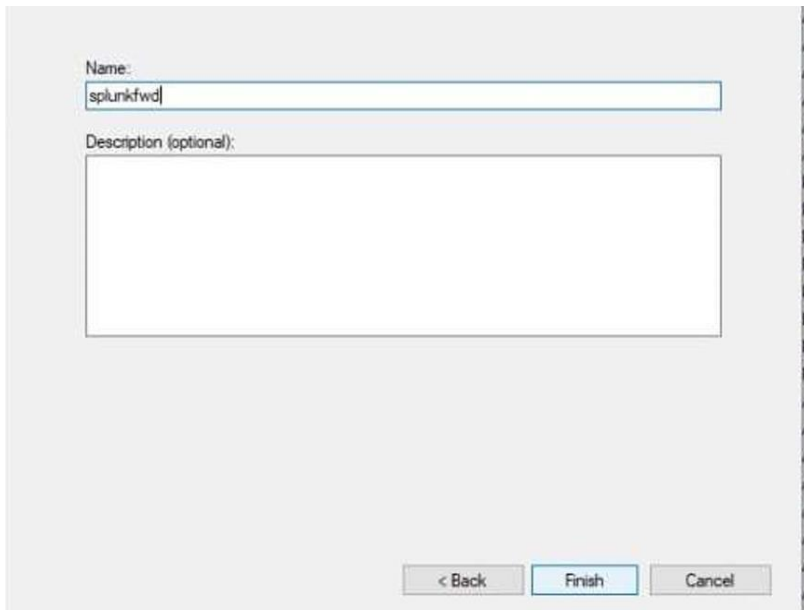
- Allow the Connection:

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



- Provide Rule Name:

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

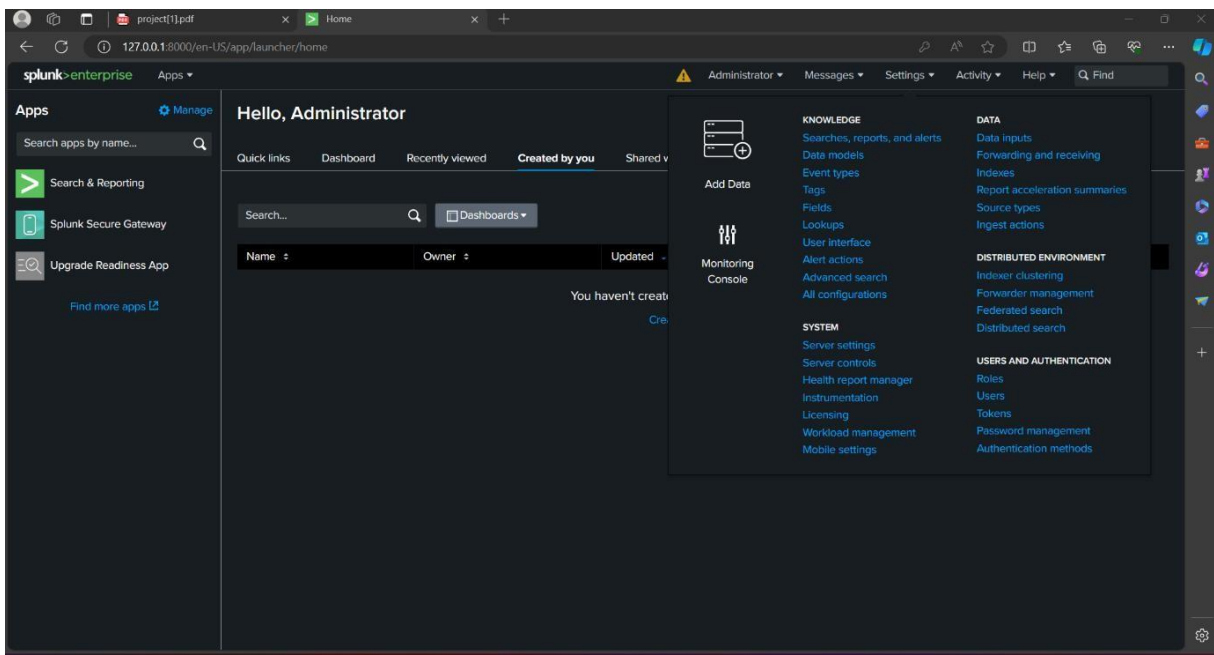


A screenshot of a web form for creating a rule. It has a "Name:" label followed by a text input field containing "splunkfwd". Below it is a "Description (optional):" label followed by a larger text area. At the bottom are three buttons: "< Back", "Finish", and "Cancel".

- Repeat for Outbound Rules:

Create "Outbound Rules" in a similar manner.

- Splunk Enterprise Configuration:
- Login to Splunk Enterprise:

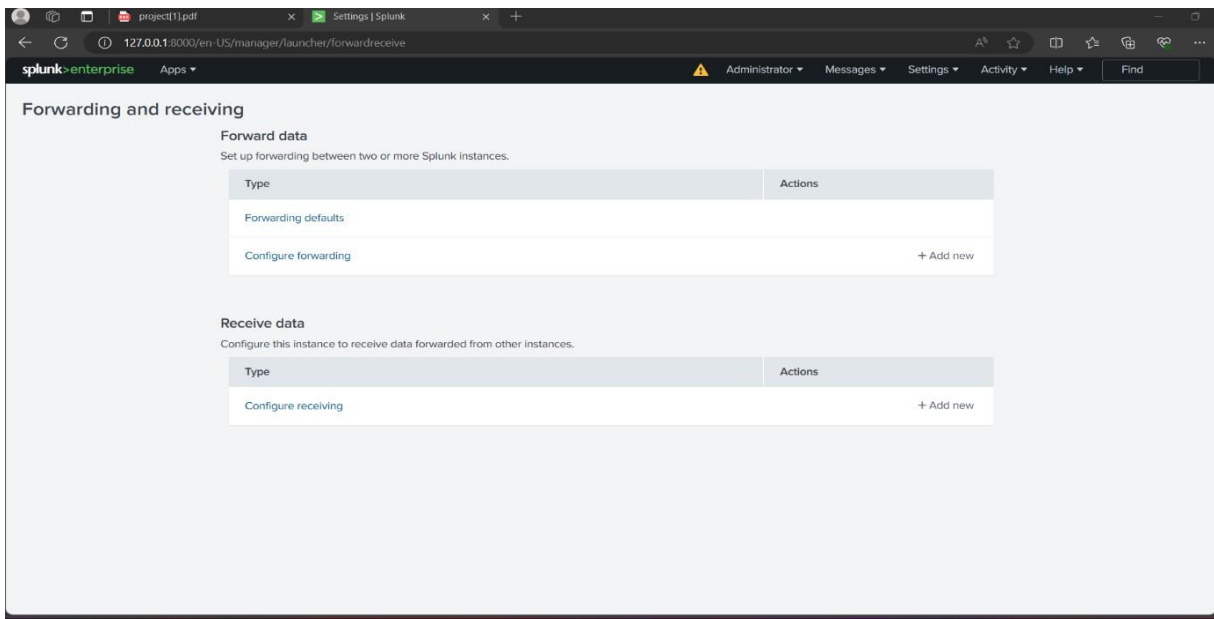


Access your Splunk Enterprise.

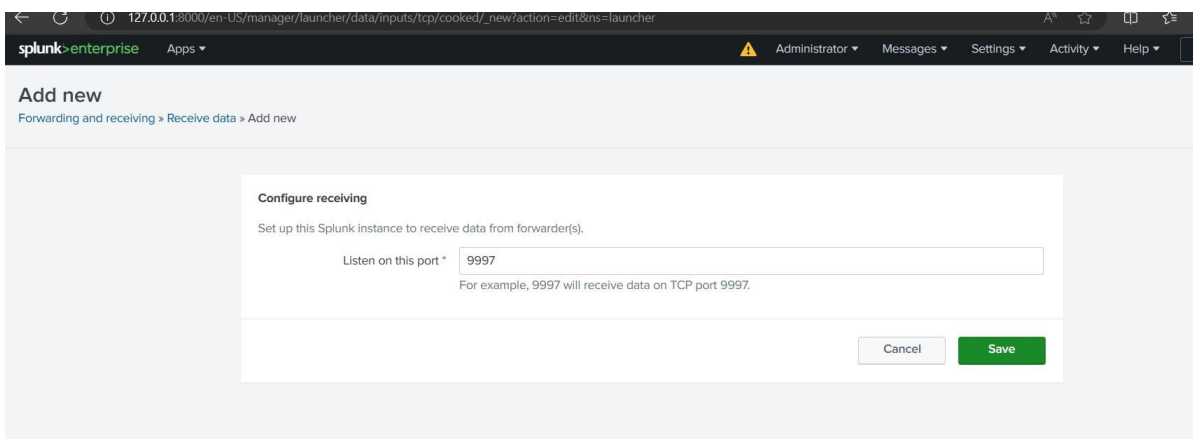
- Configure Receiving Port:

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

Click on "Configure receiving" and "New receiving port".

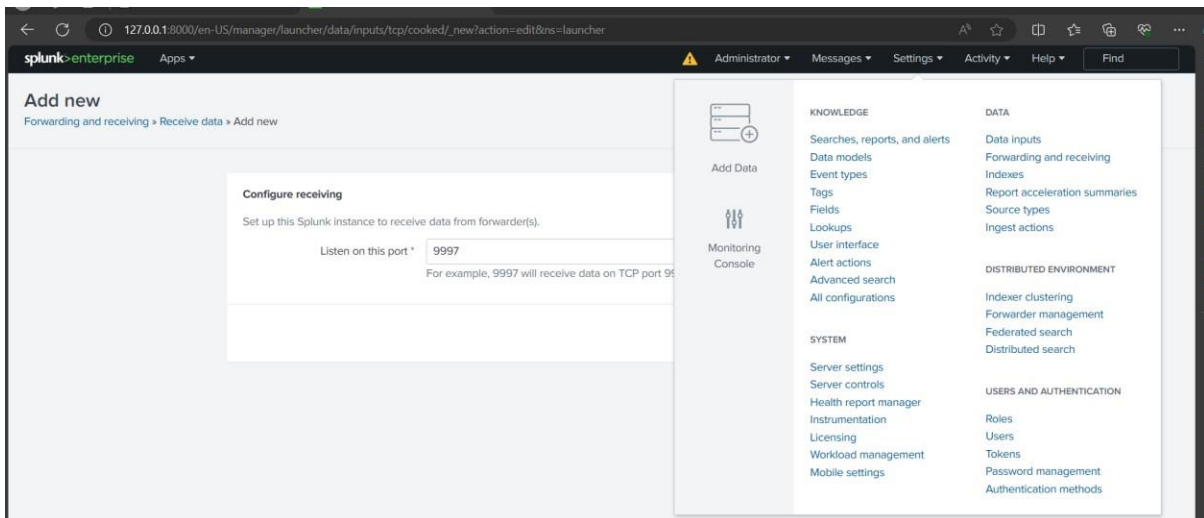


Enter the assigned before (e.g., 9997) and save.



- Configure Index in Splunk:

Go to "Settings" > "Indexes".



Ensure that the "web_logs" index is configured and enabled.

- If not, create the index:

The screenshot shows the 'Indexes' page in Splunk Enterprise. It displays a table of 16 indexes. The 'web_logs' index is not listed, indicating it needs to be created.

Name	Actions	Type	App	Current Size	Max Size	Event Count	Earliest Event	Latest Event	Home Path	Frozen Path	Status
_audit	Edit Delete Disable	Events	system	5 MB	488.28 GB	58.4K	2 days ago	a few seconds ago	\$SPLUNK_DB/auditdb	N/A	Enabled
_configtracke	Edit Delete Disable	Events	system	4 MB	488.28 GB	233	2 days ago	4 minutes ago	\$SPLUNK_DB/configtracke	N/A	Enabled
_dsappevent	Edit Delete Disable	Events	SplunkDeployment ServerConfig	1 MB	488.28 GB	0			\$SPLUNK_DB/dsappevent	N/A	Enabled
_dsclient	Edit Delete Disable	Events	SplunkDeployment ServerConfig	1 MB	488.28 GB	0			\$SPLUNK_DB/dsclient	N/A	Enabled
_dsphonehome	Edit Delete Disable	Events	SplunkDeployment ServerConfig	1 MB	488.28 GB	0			\$SPLUNK_DB/dsphonehome	N/A	Enabled
_internal	Edit Delete Disable	Events	system	33 MB	488.28 GB	481K	2 days ago	a few seconds ago	\$SPLUNK_DB/internaldb	N/A	Enabled
_introspectio	Edit Delete Disable	Events	system	88 MB	488.28 GB	44.5K	2 days ago	a few seconds ago	\$SPLUNK_DB/introspectiondb	N/A	Enabled
_metrics	Edit Delete Disable	Metrics	system	41 MB	488.28 GB	278K	2 days ago	a few seconds ago	\$SPLUNK_DB/metricsdb	N/A	Enabled
_metrics_roll	Edit Delete Disable	Metrics	system	1 MB	488.28 GB	0			\$SPLUNK_DB/metrics_rollupdb	N/A	Enabled
_telemetry	Edit Delete Disable	Events	system	1 MB	488.28 GB	42	2 days ago	3 minutes ago	\$SPLUNK_DB/telemetrydb	N/A	Enabled

Give the index name as "web_logs" and save.

New Index

General Settings

Index Name
Set index name (e.g., INDEX_NAME). Search using index=INDEX_NAME.

Index Data Type Events Metrics
The type of data to store (event-based or metrics).

Home Path
Hot/warm db path. Leave blank for default (\$SPLUNK_DB/INDEX_NAME/db).

Cold Path
Cold db path. Leave blank for default (\$SPLUNK_DB/INDEX_NAME/colddb).

Thawed Path
Thawed/resurrected db path. Leave blank for default (\$SPLUNK_DB/INDEX_NAME/thaweddb).

Data Integrity Check Enable Disable
Enable this if you want Splunk to compute hashes on every slice of your data for the purpose of data integrity.

Max Size of Entire Index GB ▾
Maximum target size of entire index.

Max Size of Hot/Warm/Cold Bucket GB ▾
Maximum target size of buckets. Enter 'auto_high_volume' for high-volume indexes.

Save Cancel

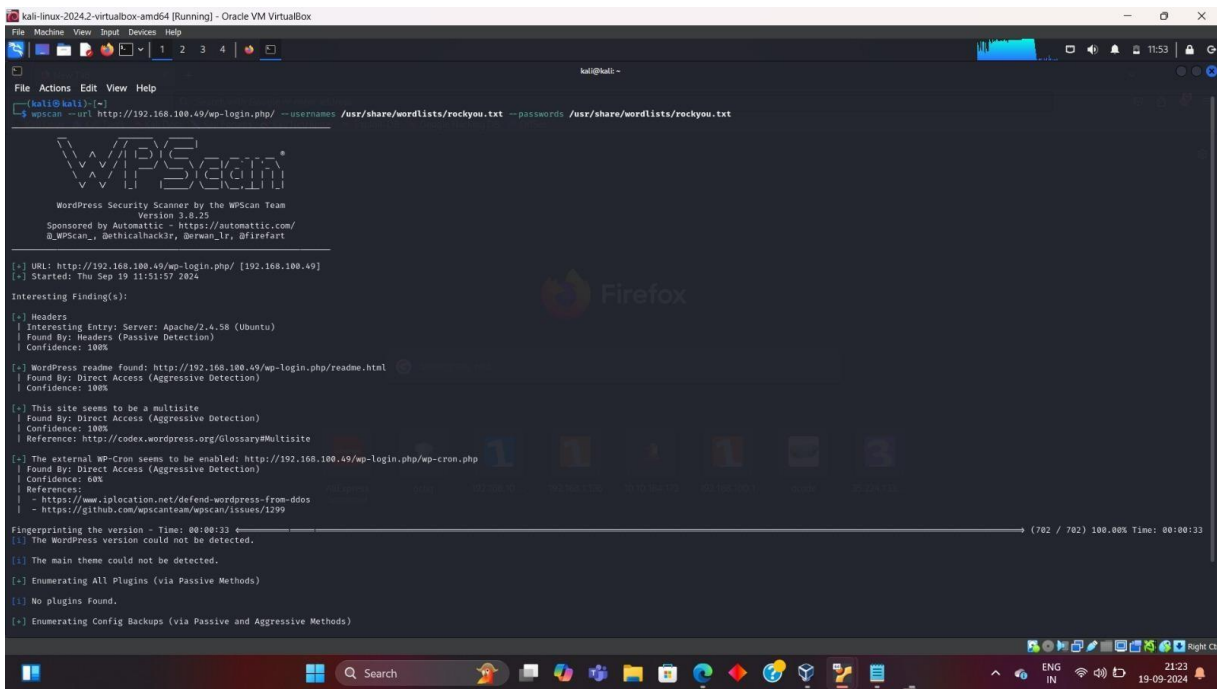
- Perform a Search in Splunk:

Navigate to “Search & Reporting.”

Search for: index=”web_logs”.

- Simulate Incorrect Login Events:

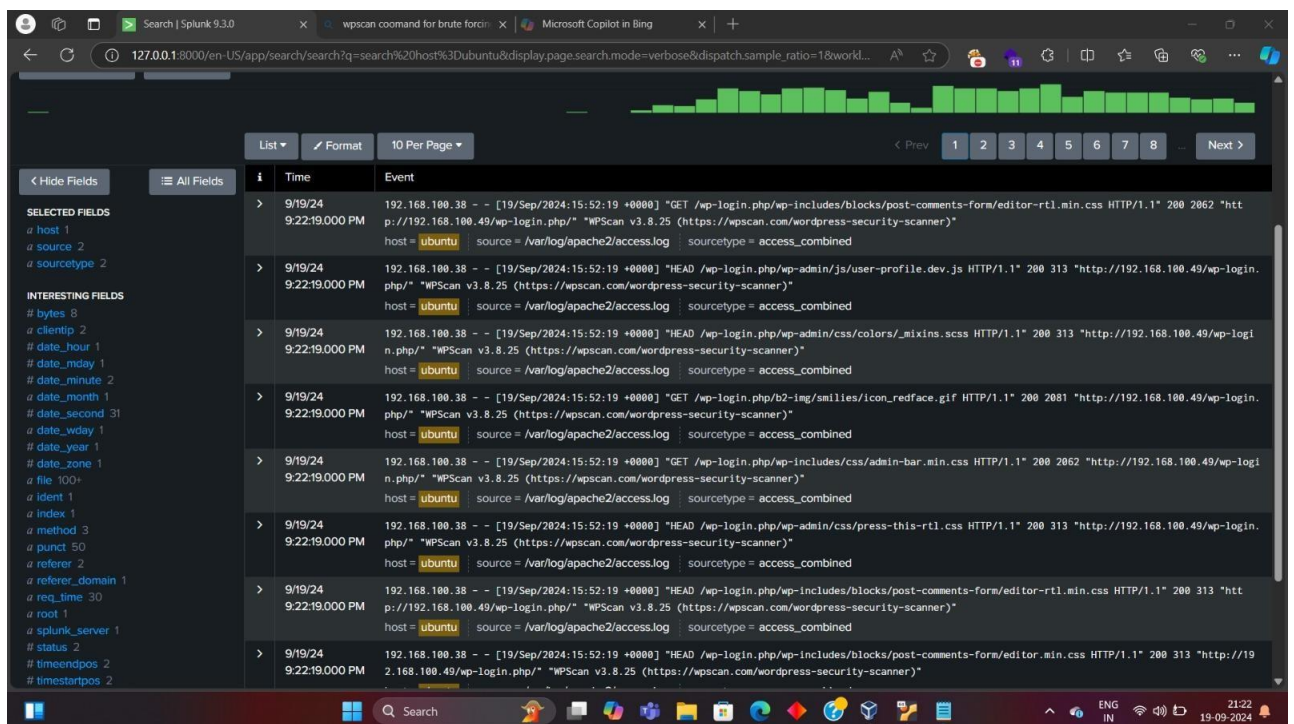
I have done brute forcing using wpscan in the wordpress admin login



Perform incorrect login events by bruteforcing using the command :

[“wpscan --url http://ip/wordpress/wp-login.php/ --usernames /usr/share/wordlists/rockyou.txt --passwords /usr/share/wordlists/rockyou.txt”](#)

- Refresh Splunk



- Verify events in Splunk.
 - Confirm that the simulated incorrect login events are visible in Spunk search results as in the above picture.
-

PROJECT SUMMARY :

The project involves deploying WordPress on an Ubuntu 23.10 server and implementing website log monitoring for effective website management.

1. Deployment of WordPress on Ubuntu 23.10:

- Install Ubuntu 23.10 on a server or virtual machine.
- Set up LAMP (Linux, Apache, MySQL, PHP) stack on the Ubuntu server
- Download and configure WordPress files on the server.
- Create a MySQL database for WordPress and configure it.
- Configure Apache to serve WordPress site.

2. Implementing Website Log Monitoring:

- Configure log collection from Apache server to gather website access logs.
- Set up log parsing and filtering to extract relevant information such as HTTP status codes, request URLs, client IPs, etc.
- Set up alerts for abnormal activities or errors detected in website logs.

3. Security and Performance Optimization:

- Implement security best practices for WordPress, such as using strong passwords, updating plugins and themes regularly, and implementing security plugins.

4. Documentation and Training:

- Document the deployment process, configuration settings, and monitoring procedures for future reference.

5. Maintenance and Support:

- Establish a maintenance schedule for regular updates and backups of the WordPress site and log monitoring system.

- Provide ongoing support to troubleshoot issues and optimize system performance.

By deploying WordPress on Ubuntu 23.10 and implementing website log monitoring, the project aims to enhance the security, performance, and management of the website, ensuring a seamless user experience and proactive detection of any issues.
