How To Install Wordpress On Ubuntu 22.04 and Monitor Website Logs

In this comprehensive guide, we'll walk you through the step-by-step process of installing WordPress on Ubuntu 22.04, ensuring a seamless setup for your website. But that's not all - we'll also delve into the crucial aspect of monitoring website logs, empowering you with the tools to track performance, troubleshoot issues, and keep your WordPress site running smoothly. Let's dive in and simplify the process of both installation and monitoring for a robust online presence. Here we use ubuntu server and install apache to host WordPress site and Splunk to monitor the logs from it.

STEP 1: Ubuntu Server Installation on VirtualBox

Download Ubuntu Server 22.04 from the link given below. Install and configure a virtual machine in VirtualBox.

Link: https://ubuntu.com/download/server

STEP 2: Install Apache

Open the terminal on your Ubuntu system. The terminal is a text interface to your computer, which you will use to run all the commands.

First, update your software package list.

```
ubuntu@ubuntu:~$ sudo apt-get update
```

If you encounter any issues with missing tools, run this command.

```
ubuntu@ubuntu:~$ sudo apt install net-tools
```

Run the below command to install Apache 2 on Ubuntu 22.04.

```
ubuntu@ubuntu:~$ sudo apt install apache2
```

It is necessary to allow Apache2 to start at the system boot time and to start the service to verify its status as well.

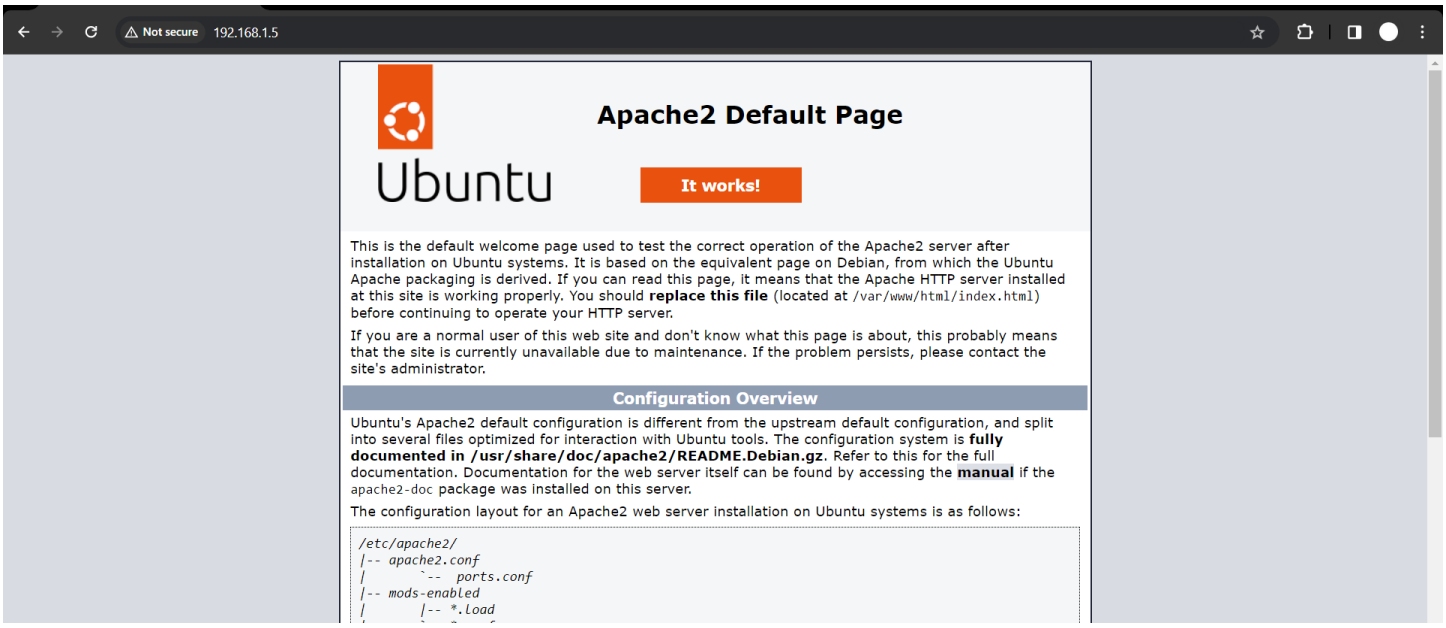
```
ubuntu@ubuntu:~$ sudo systemctl enable apache2
```

```
ubuntu@ubuntu:~$ sudo systemctl status apache2
```

```
umesh@umesh:/var/www/html$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2024-01-10 18:33:37 UTC; 7min ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 27245 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 27249 (apache2)
      Tasks: 6 (limit: 2221)
     Memory: 13.9M
        CPU: 65ms
     CGroup: /system.slice/apache2.service
             ├─27249 /usr/sbin/apache2 -k start
             ├─27250 /usr/sbin/apache2 -k start
             ├─27251 /usr/sbin/apache2 -k start
             ├─27252 /usr/sbin/apache2 -k start
             ├─27253 /usr/sbin/apache2 -k start
             └─27254 /usr/sbin/apache2 -k start

Jan 10 18:33:37 umesh systemd[1]: apache2.service: Deactivated successfully.
Jan 10 18:33:37 umesh systemd[1]: Stopped The Apache HTTP Server.
Jan 10 18:33:37 umesh systemd[1]: Starting The Apache HTTP Server...
Jan 10 18:33:37 umesh apachectl[27248]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to
Jan 10 18:33:37 umesh systemd[1]: Started The Apache HTTP Server.
```

Open your web browser, and search your ip in the address box to verify that the **Apache** server has been started.

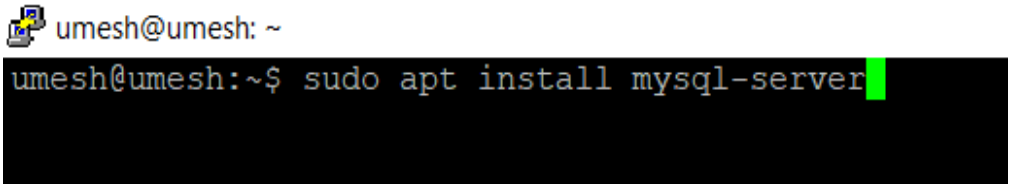
If the Apache2 web server is running, it will display the default Apache2 index page.



STEP 3: Install MySQL

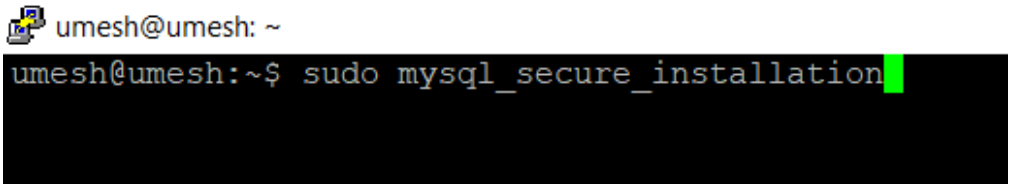
After Apache has been started, it is time to install MySQL. Run the following command in the terminal to do this:

```
ubuntu@ubuntu:~$ sudo apt install mysql-server
```



It is highly recommended that you run a security program after the database server has been installed to remove unsecure default settings and protect your database.

```
ubuntu@ubuntu:~$ sudo mysql_secure_installation
```



You will be asked to install the **validate_password** plugin. So, type Y/Yes, then press **Enter** and finally choose the default password strength.

To answer the remaining questions, press Y and hit the **ENTER** key for each prompt.

This command will also enable MySQL to begin on boot.

```
ubuntu@ubuntu:~$ sudo systemctl enable mysql
```

```
ubuntu@ubuntu:~$ sudo systemctl status mysql
```

STEP 4: Install PHP

WordPress is a PHP-based CMS. We need PHP to process the dynamic content on our WordPress site.

Ubuntu 20.04 defaults to PHP 7.4. We will need additional modules to allow PHP to communicate with Apache and MySQL instances. The following command will install PHP along with the MySQL and Apache modules:

ubuntu@ubuntu:~$ sudo apt install php libapache2-mod-php php-mysql

umesh@umesh: ~

```
umesh@umesh:~$ sudo apt install php libapache2-mod-php php-mysql
```

WordPress and many plugins use PHP extensions, which you will need to install manually.

ubuntu@ubuntu:~$ sudo apt install php-curl php-gd php-mbstring php-xml php-xmlrpc php-soap php-intl php-zip

umesh@umesh: ~

```
umesh@umesh:~$ sudo apt install php-curl php-gd php-mbstring php-xml php-xmlrpc php-soap php-intl php-zip
```

The following command will verify that PHP 7.4 has been successfully installed:

```
ubuntu@ubunu:$ php -v
```

```
umesh@umesh:~$ php -v
PHP 8.1.2-1ubuntu2.14 (cli) (built: Aug 18 2023 11:41:11) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.1.2, Copyright (c) Zend Technologies
    with Zend OPcache v8.1.2-1ubuntu2.14, Copyright (c), by Zend Technologies
umesh@umesh:~$ _
```

After PHP has been installed and any required extensions have been installed, Apache must be restarted to load the new extensions.

ubuntu@ubuntu:~$ sudo systemctl restart apache2

STEP 5: Install WordPress

First, we will download the WordPress installation files and place them in the default web server root directory /var/www/html.

ubuntu@ubuntu:~$ cd /var/www/html

Now download the latest WordPress install with the following command.

ubuntu@ubuntu:~$/var/www/html$ sudo wget -c http://wordpress.org/latest.tar.gz

umesh@umesh: /var/www/html

```
umesh@umesh:/var/www/html$ sudo wget -c http://wordpress.org/latest.tar.gz
```

Extract the files

ubuntu@ubuntu: ~$/var/www/html$ sudo tar -xzvf latest.tar.gz

ubuntu@ubuntu: ~$/var/www/html$ ls -l

umesh@umesh: /var/www/html

```
umesh@umesh:/var/www/html$ ls -l
total 23924
-rw-r--r-- 1 root     root        10671 Jan 10 18:25 index.html
-rw-r--r-- 1 root     root     24479697 Dec  6 16:26 latest.tar.gz
drwxr-xr-x 5 www-data www-data     4096 Dec  6 16:25 wordpress
umesh@umesh:/var/www/html$
```

The extracted WordPress files will be now placed in the WordPress directory at the following location on your server

/var/www/html/wordpress

The user of your web server must own these files.

We are using Apache as our web server. Apache is running on Ubuntu 20.04. The following command will allow you to change the owner of these files and set the appropriate permissions:

ubuntu@ubuntu: ~$ sudo chown -R www-data:www-data /var/www/html/wordpress

STEP 6: Create a Database for WordPress

Next, we will create a WordPress database for the site and set up a user account. This will make it easier to manage the site and increase its security.

Log in to your MySQL root account via Terminal by entering:

ubuntu@ubuntu: ~$/var/www/html$ sudo mysql -u root -p

Create a separate database for WordPress to manage

```
mysql>CREATE DATABASE demo_db;
```

To access the new database, we will create a MySQL user account. Enter a strong password

mysql>CREATE USER ‘demo_user’@’%’ IDENTIFIED BY 'demo-password';

You have just created a new user. Next, let the database know that your **demo_user** should have complete access to the database you set up:

```
mysql>GRANT ALL ON demo_db.* TO 'demo_user'@'%';
```

You now have a database and user account, each made specifically for WordPress. You need to flush the privileges so that the current instance of MySQL knows about the recent changes made:

```
mysql>FLUSH PRIVILEGES;
```

Exit out of MySQL by writing the following:

```
mysql>exit;
```

umesh@umesh: /var/www/html

```
umesh@umesh:/var/www/html$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 8.0.35-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2023, 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> CREATE DATABASE umesh;
Query OK, 1 row affected (0.01 sec)

mysql> CREATE USER 'umesh'@'%' IDENTIFIED BY 'tezlo@123';
ERROR 1819 (HY000): Your password does not satisfy the current policy requiremen
ts
mysql> CREATE USER 'umesh'@'%' IDENTIFIED BY 'Tezlo@123';
Query OK, 0 rows affected (0.01 sec)

mysql> GRANT ALL ON umesh.* TO 'umesh'@'%';
Query OK, 0 rows affected (0.01 sec)

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

mysql> EXIT
Bye
umesh@umesh:/var/www/html$
```

Allow the executable permission to be granted to the WordPress folder.

ubuntu@ubuntu: ~$/var/www/html$ sudo chmod -R 777 wordpress/

ubuntu@ubuntu: ~$/var/www/html$ cd wordpress/

STEP 7: Setup and Configure Wordpress

After setting up a database for WordPress, the next and final step is to set up and configure WordPress. Firstly, you need to create a configuration file for WordPress. So, rename the sample WordPress configuration file using the following command:

ubuntu@ubuntu: ~$/var/www/html/wordpress$ mv wp-config-sample.php wp-config.php

Edit the wpconfig. As shown below, edit the php file.

ubuntu@ubuntu: ~$/var/www/html/wordpress$ sudo nano wp-config.php

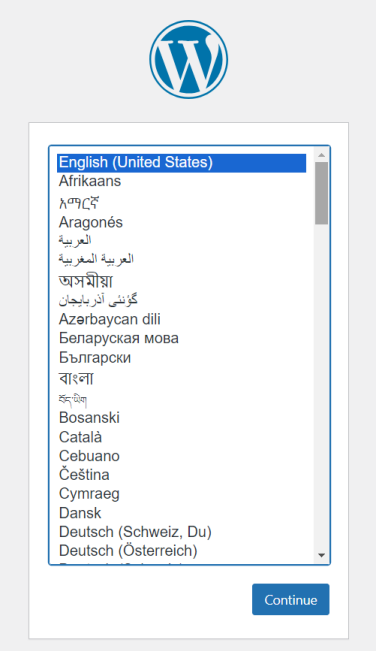
Update the database settings by replacing demo_db, demo_user, and demo_password with your own details.

```
<?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 web site, 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://wordpress.org/documentation/article/editing-wp-config-php/
 *
 * @package WordPress
 */

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

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

/** Database password */
define( 'DB_PASSWORD', 'Tezlo@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}.
 *
 * You can change these at any point in time to invalidate all existing cookies.

^G Help      ^O Write Out   ^W Where Is    ^K Cut       ^T Execute   ^C Location  M-U Undo      M-A
```

Save the file and close it.

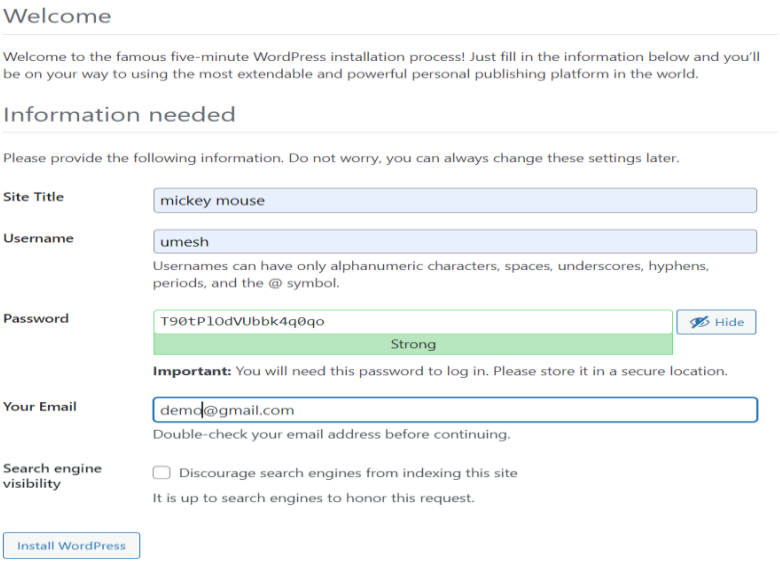
Once you have done this, you can access your WordPress page to finish the installation. Open the browser and go to : https://your_server_IP/wordpress/

The next screen will open. Click on Continue to select the language.

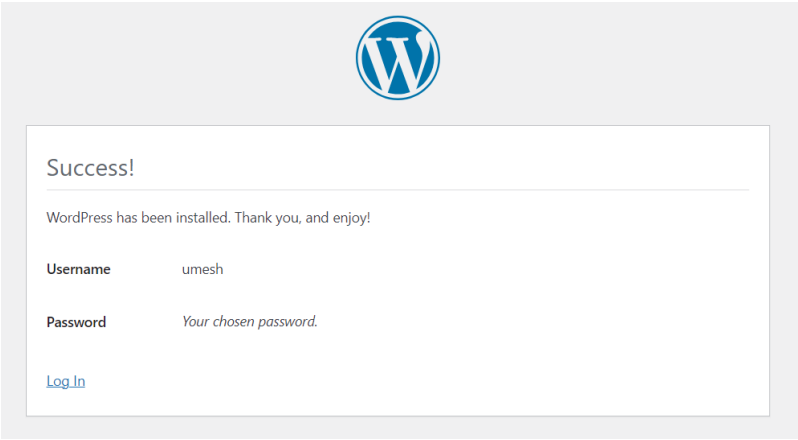


Click on “Install WordPress” to enter your preferred information, including site title, username, and password.

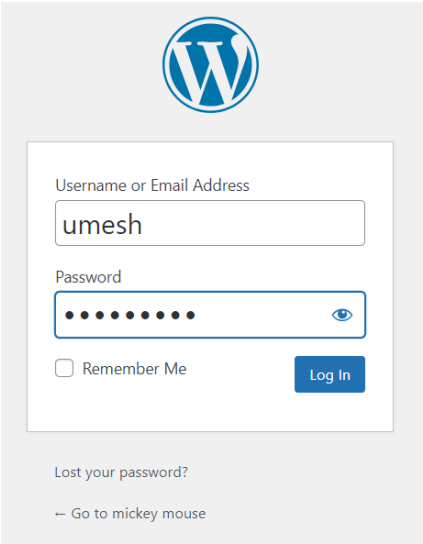
- **Site Title:** Enter the WordPress website name. We recommend entering the domain name to optimize your site.
- **Username:** Create a new username to log in to WordPress.
- **Password:** Create a password to protect your WordPress account.
- **Your email**: Add your email address to receive updates and notifications.
- **Search engine visibility:** You can leave this box unchecked to prevent search engines from indexing your site until it’s ready.



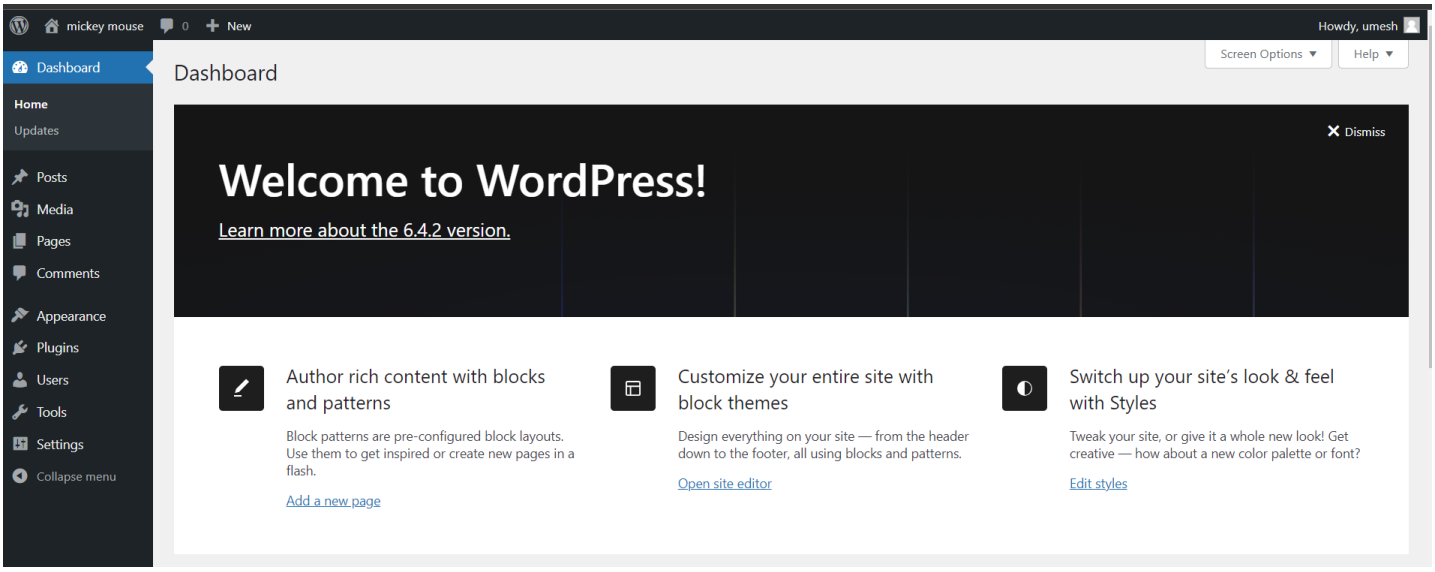
WordPress will now be installed successfully. You can log in to your admin dashboard with the previously set up information.



To log in, enter your username and password.



After successfully logging in, the WordPress dashboard page will greet you.



WordPress Site Loading Problem

If you've changed the network and your resulting IP address is different, you may encounter issues accessing your WordPress site. Here's a step-by-step guide to troubleshoot and resolve the issue:

Update wp-config.php

Access the WordPress `wp-config.php` file located in your WordPress installation directory /var/www/html/wordpress/wp-config.php

```
ubuntu@ubuntu: ~$ sudo nano wp-config.php
```

Add the `WP_HOME` and `WP_SITEURL` constants with the new IP address above the line

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

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

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

Replace `'your-new-ip'` with the actual new IP address.

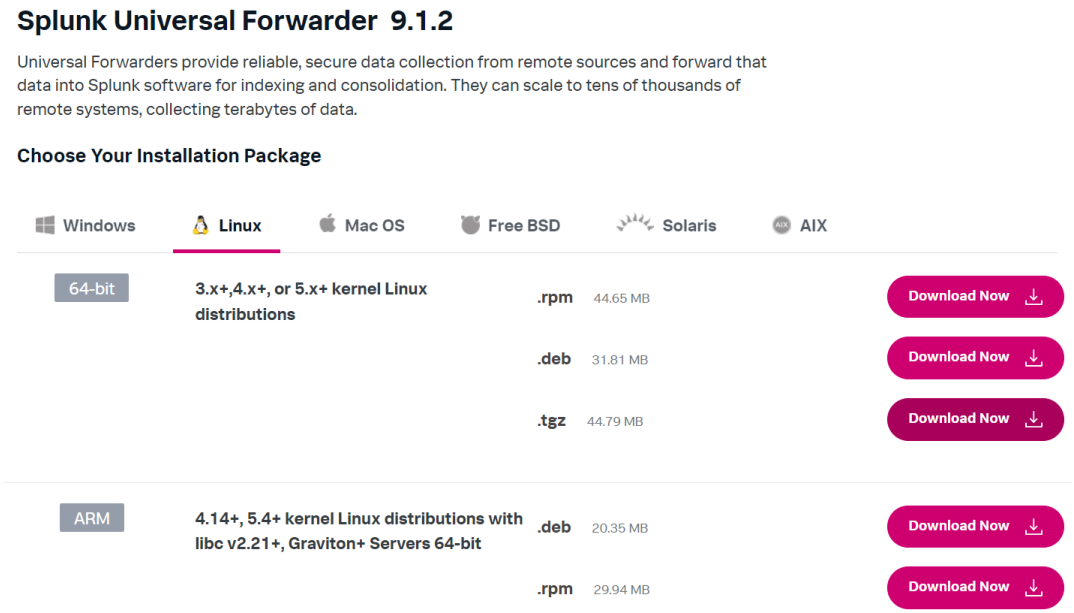
```
define( 'LOGGED_IN_KEY',    'put your unique phrase here' );
define( 'NONCE_KEY',        'put your unique phrase here' );
define( 'AUTH_SALT',        'put your unique phrase here' );
define( 'SECURE_AUTH_SALT', 'put your unique phrase here' );
define( 'LOGGED_IN_SALT',   'put your unique phrase here' );
define( 'NONCE_SALT',       'put your unique phrase here' );

/**#@-*/

/**
 * WordPress database table prefix.
 *
 * You can have multiple installations in one database if you give each
 * a unique prefix. Only numbers, letters, and underscores please!
 */
$table_prefix = 'wp_';

/**
 * For developers: WordPress debugging mode.
 *
 * Change this to true to enable the display of notices during development.
 * It is strongly recommended that plugin and theme developers use WP_DEBUG
 * in their development environments.
 *
 * For information on other constants that can be used for debugging,
 * visit the documentation.
 *
 * @link https://wordpress.org/documentation/article/debugging-in-wordpress/
 */
define( 'WP_DEBUG', false );

/* Add any custom values between this line and the "stop editing" line. */

define('WP_HOME', 'http://192.168.0.233/wordpress');
define('WP_SITEURL', 'http://192.168.0.233/wordpress');

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

/** Absolute path to the WordPress directory. */
if ( ! defined( 'ABSPATH' ) ) {
        define( 'ABSPATH', __DIR__ . '/' );
}

/** Sets up WordPress vars and included files. */
require_once ABSPATH . 'wp-settings.php';

^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute
```

After making changes, restart Apache

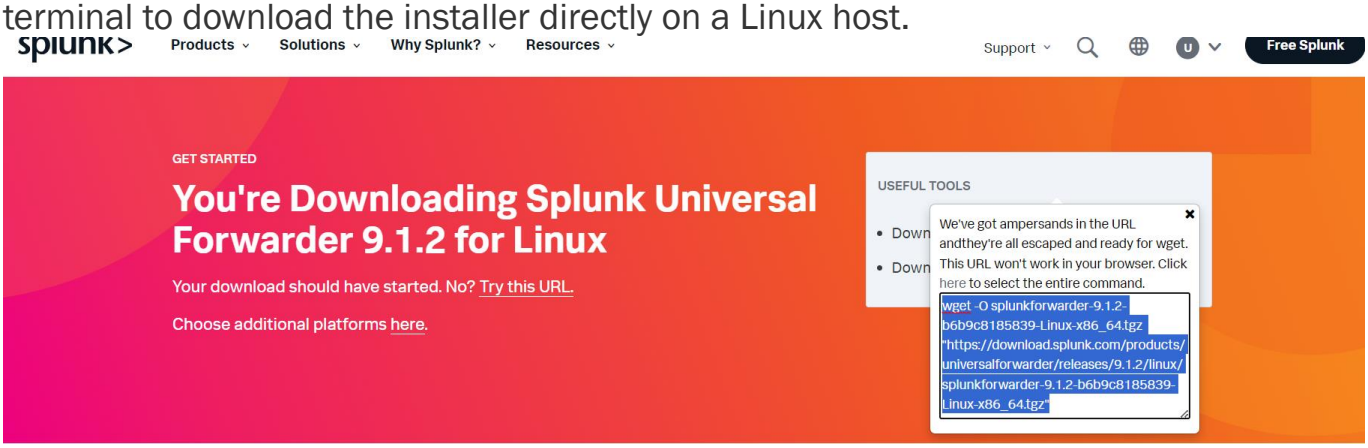
```
ubuntu@ubuntu: ~$sudo systemctl restart apache2
```

STEP 8: Deploying the Splunk Universal forwarder on Ubuntu

You will need a Splunk.com account to access the download. After login, click on “free splunk” then click on “Free trails and Downloads page” and scroll down there you will see Universal Forwarder (get my free download). There you”ll want to click on the Linux tab and choose the download package (choose the correct file 32/64bit) (.tgz).



Clicking the download link will take you to a page that automatically downloads the installer of choice. One useful tool (conveniently placed in the “useful tools” section) is the “download via command line” option, which gives you a wget link that can be pasted into a terminal to download the installer directly on a Linux host.



Start by downloading the .tgz installer from Splunk on /tmp directory by using the link.

ubuntu@ubuntu: ~$ cd /tmp

ubuntu@ubuntu:/tmp~$ <copied_link>

```
umesh@umesh:~$ cd /tmp
umesh@umesh:/tmp$ wget -O splunkforwarder-9.1.2-b6b9c8185839-Linux-x86_64.tgz "https://download.splunk.com/products/universalforwarder/releases/9.1.2/linux/splunkforwarder-9.1.2-b6b9c818583
9-Linux-x86_64.tgz"
--2024-01-11 02:22:57--  https://download.splunk.com/products/universalforwarder/releases/9.1.2/linux/splunkforwarder-9.1.2-b6b9c8185839-Linux-x86_64.tgz
Resolving download.splunk.com (download.splunk.com)... 65.8.76.33, 65.8.76.19, 65.8.76.122, ...
Connecting to download.splunk.com (download.splunk.com)|65.8.76.33|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 46963654 (45M) [binary/octet-stream]
Saving to: 'splunkforwarder-9.1.2-b6b9c8185839-Linux-x86_64.tgz'

splunkforwarder-9.1.2-b6b9c8185839-Linux-x86_64  63%[=============================================================>                          ]  28.53M  3.00MB/s    eta 8s
```

Then, extract the .tgz file to the location where you want to run the Universal Forwarder. Generally, pick the default location of /opt/splunkforwarder. The following commands can be used to accomplish this (assuming that the UF package is downloaded to /tmp)

ubuntu@ubuntu:/tmp~$ sudo tar xvzf splunkforwarder-9.1.2-b6b9c8185839-Linux-x86_64.tgz -C /opt

Note: most versions of tar support the -C argument to specify a directory to extract the tarball. If your version of tar doesn't support this argument, you can also switch into the directory where you're looking to extract the UF package.

```
umesh@umesh:/tmp$
umesh@umesh:/tmp$ ls
snap-private-tmp                                                    systemd-private-12bbce041adc499d9b92829e64e565b3-systemd-logind.service-Ux1arD
splunkforwarder-9.1.2-b6b9c8185839-Linux-x86_64.tgz                 systemd-private-12bbce041adc499d9b92829e64e565b3-systemd-resolved.service-OQBJMw
systemd-private-12bbce041adc499d9b92829e64e565b3-apache2.service-boyo4C   systemd-private-12bbce041adc499d9b92829e64e565b3-systemd-timesyncd.service-JWzwPR
systemd-private-12bbce041adc499d9b92829e64e565b3-ModemManager.service-YWQVnp
umesh@umesh:/tmp$ sudo tar xvzf splunkforwarder-9.1.2-b6b9c8185839-Linux-x86_64.tgz -C /opt
[sudo] password for umesh:
splunkforwarder/
splunkforwarder/swidtag/
splunkforwarder/swidtag/splunk-UniversalForwarder-primary.swidtag
splunkforwarder/ftr
splunkforwarder/openssl/
splunkforwarder/openssl/misc/
splunkforwarder/openssl/misc/c_info
splunkforwarder/openssl/misc/tsget
splunkforwarder/openssl/misc/c_issuer
splunkforwarder/openssl/misc/CA.sh
splunkforwarder/openssl/misc/c_hash
splunkforwarder/openssl/misc/c_name
splunkforwarder/openssl/misc/CA.pl
splunkforwarder/openssl/openssl.cnf
splunkforwarder/openssl/copyright.txt
splunkforwarder/share/
splunkforwarder/share/mongo-c-driver/
splunkforwarder/share/mongo-c-driver/uninstall.sh
splunkforwarder/share/mongo-c-driver/NEWS
splunkforwarder/share/mongo-c-driver/COPYING
splunkforwarder/share/mongo-c-driver/README.rst
splunkforwarder/share/mongo-c-driver/THIRD_PARTY_NOTICES
splunkforwarder/share/copyright.txt
splunkforwarder/share/splunk/
splunkforwarder/share/splunk/3rdparty/
splunkforwarder/share/splunk/3rdparty/Copyright-for-zc.lockfile-2.0.txt
splunkforwarder/share/splunk/3rdparty/Copyright-for-pcre2-10.40.txt
splunkforwarder/share/splunk/3rdparty/Copyright-for-jemalloc-4.5.0.txt
splunkforwarder/share/splunk/3rdparty/Copyright-for-google-cloud-cpp-1.14.0.txt
splunkforwarder/share/splunk/3rdparty/Copyright-for-prometheus-cpp-0.9.0.txt
splunkforwarder/share/splunk/3rdparty/Copyright-for-setuptools_scm_git_archive-1.1.txt
splunkforwarder/share/splunk/3rdparty/Copyright-for-FormEncode-1.3.1.txt
splunkforwarder/share/splunk/3rdparty/Copyright-for-double-conversion-3.0.0.txt
splunkforwarder/share/splunk/3rdparty/Copyright-for-Babel-2.9.1.txt
splunkforwarder/share/splunk/3rdparty/Copyright-for-snappy-1.1.8.txt
splunkforwarder/share/splunk/3rdparty/Copyright-for-decorator-4.4.0.txt
splunkforwarder/share/splunk/3rdparty/Copyright-for-CherryPy-18.1.2.txt
```

Next, to start the Splunk Forwarder, navigate to bin directory on Splunk Forwarder:

ubuntu@ubuntu:~$ cd /opt/splunkforwarder/bin

Next, start the Splunk Forwarder:

ubuntu@ubuntu:/opt/splunkforwarder/bin ~$ sudo ./splunk start --accept-license

You'll be prompted to specify an administrator username and password; this is the account that was specified above for troubleshooting the UF. It does not (and should not) need to be an account that already exists on the system.

```
umesh@umesh:/opt/splunkforwarder/bin$ sudo su
root@umesh:/opt/splunkforwarder/bin# ./splunk start --accept-license

This appears to be your first time running this version of Splunk.

Splunk software must create an administrator account during startup. Otherwise, you cannot log in.
Create credentials for the administrator account.
Characters do not appear on the screen when you type in credentials.

Please enter an administrator username: umesh
Password must contain at least:
   * 8 total printable ASCII character(s).
Please enter a new password:
Please confirm new password:
Creating unit file...
Failed to auto-set default user.
Failed to create the unit file. Please do it manually later.


Splunk> Like an F-18, bro.

Checking prerequisites...
        Checking mgmt port [8089]: open
                Creating: /opt/splunkforwarder/var/lib/splunk
                Creating: /opt/splunkforwarder/var/run/splunk/appserver/i18n
                Creating: /opt/splunkforwarder/var/run/splunk/appserver/modules/static/css
                Creating: /opt/splunkforwarder/var/run/splunk/upload
                Creating: /opt/splunkforwarder/var/run/splunk/search_telemetry
                Creating: /opt/splunkforwarder/var/run/splunk/search_log
                Creating: /opt/splunkforwarder/var/spool/splunk
                Creating: /opt/splunkforwarder/var/spool/dirmoncache
                Creating: /opt/splunkforwarder/var/lib/splunk/authDb
                Creating: /opt/splunkforwarder/var/lib/splunk/hashDb
New certs have been generated in '/opt/splunkforwarder/etc/auth'.
        Checking conf files for problems...
        Done
        Checking default conf files for edits...
        Validating installed files against hashes from '/opt/splunkforwarder/splunkforwarder-9.1.2-b6b9c8185839-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 set to "1" fo
r increased security
Done

root@umesh:/opt/splunkforwarder/bin#
```

Finally, enable the Universal Forwarder to start on boot:

ubuntu@ubuntu:/opt/splunkforwarder/bin ~$ sudo ./splunk enable boot-start

```
root@umesh:/opt/splunkforwarder/bin# ./splunk enable boot-start
splunk is currently running, please stop it before running enable/disable boot-start
root@umesh:/opt/splunkforwarder/bin#
```

At this point, the Universal Forwarder installation is complete. Now we need to Configure the Deployment Server.

STEP 9: Configuring Splunk Universal Forwarder

Configuring Splunk Universal Forwarder on an Ubuntu server to monitor logs from another Windows machine involves a few steps.

Open the **output.conf** file for editing. This file is located in the Splunk Universal Forwarder configuration directory. The default path is /opt/splunkforwarder/etc/system/local/

(if there no **outputs.conf** file on /opt/splunkforwarder/etc/system/local/ then create one using the command ubuntu@ubuntu:/opt/splunkforwarder/etc/system/local/ ~$ sudo nano outputs.conf)

```
umesh@umesh:/opt/splunkforwarder/etc/system/local$ ls
README  server.conf
umesh@umesh:/opt/splunkforwarder/etc/system/local$ sudo nano outputs.conf
```

Add the following lines to the **outputs.conf** file:

[tcpout]

defaultGroup = splunk-group

[tcpout:splunk-group]

server = <splunk_server_ip>:<splunk_listener_port>

***Remember** to replace **<splunk_server_ip>** (where you want to send your logs to, in my case its my windows machine ip) and **<splunk_listener_port>** (eg:9997) with the appropriate values for your Splunk server.

```
  GNU nano 6.2                          outputs.conf
[tcpout]
defaultGroup = splunk-group

[tcpout:splunk-group]
server = 192.168.1.5:9995
```

Now, you need to configure which logs to forward. This is done using the **inputs.conf** file.

Open the **inputs.conf** file for editing. This file is located in the Splunk Universal Forwarder configuration directory. The default path is **/opt/splunkforwarder/etc/system/local/inputs.conf**.

(if there no **inputs.conf** file on /opt/splunkforwarder/etc/system/local/ then create one using the command ubuntu@ubuntu:/opt/splunkforwarder/etc/system/local/ ~$ sudo nano inputs.conf)

```
umesh@umesh:/opt/splunkforwarder/etc/system/local$ ls
outputs.conf  README  server.conf
umesh@umesh:/opt/splunkforwarder/etc/system/local$ sudo nano inputs.conf
```

Add configurations for the logs you want to monitor. For example, to monitor Apache access logs. (You can view available logs on /var/logs/apache2)

[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

```
  GNU nano 6.2                          inputs.conf
[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
```

Save the file and restart the Splunk Universal Forwarder:

ubuntu@ubuntu: ~$ sudo /opt/splunkforwarder/bin/splunk restart

Adjusting the firewall settings

you may need to adjust the firewall settings to allow communication between the Splunk Universal Forwarder on your Ubuntu server and the Splunk server.

Check Current Firewall Status: ubuntu@ubuntu: ~$sudo ufw status

If firewall inactive: ubuntu@ubuntu: ~$sudo ufw enable

Allow Splunk Forwarder Traffic:

ubuntu@ubuntu: ~$ sudo ufw allow <splunk_listener_port>/tcp

Replace **<splunk_listener_port>** with the port you want to sent log to (eg:9997)

```
umesh@umesh:~$ sudo ufw allow 9995/tcp
Rule added
Rule added (v6)
umesh@umesh:~$
```

{ It is better to allow these ports to avoid connection errors (80/tcp, 22, 443/tcp)
ubuntu@ubuntu: ~$sudo ufw allow 80/tcp
ubuntu@ubuntu: ~$sudo ufw allow 22
ubuntu@ubuntu: ~$sudo ufw allow 443/tcp }

Reload the firewall:

ubuntu@ubuntu: ~$sudo ufw reload

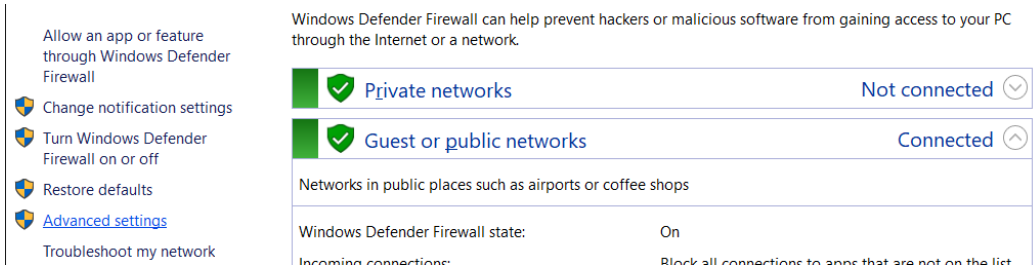
Restart Splunk Universal Forwarder:

ubuntu@ubuntu: ~$ sudo /opt/splunkforwarder/bin/splunk restart

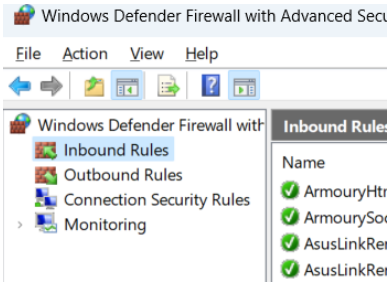
STEP 10: Configuring Splunk Server

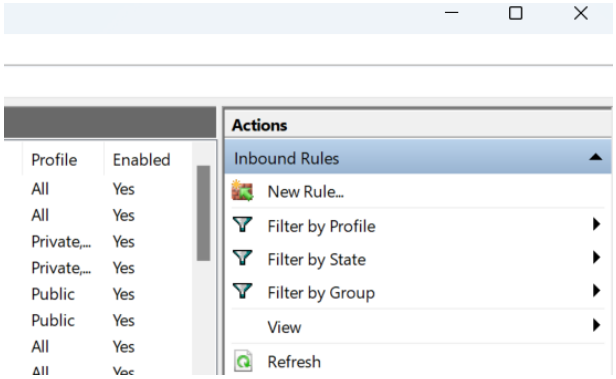
If your Splunk server is running on another machine (e.g., a Windows machine), you need to ensure that the Windows Firewall or any other firewall software is configured to allow incoming traffic on **<splunk_listener_port>** you assigned before(eg:9997).

Go to windows settings> Windows Defender Firewall. Click on "Advanced settings" on the left panel.

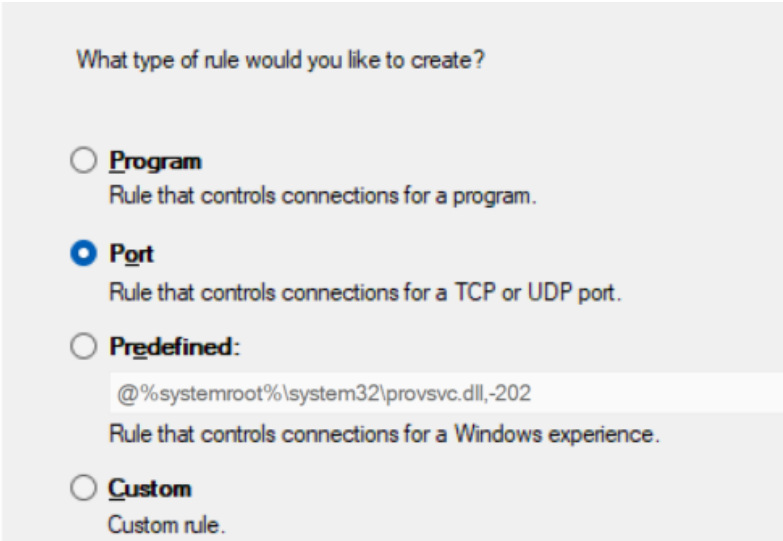


In the Windows Firewall with Advanced Security window, right-click on "Inbound Rules" and choose "New Rule..."

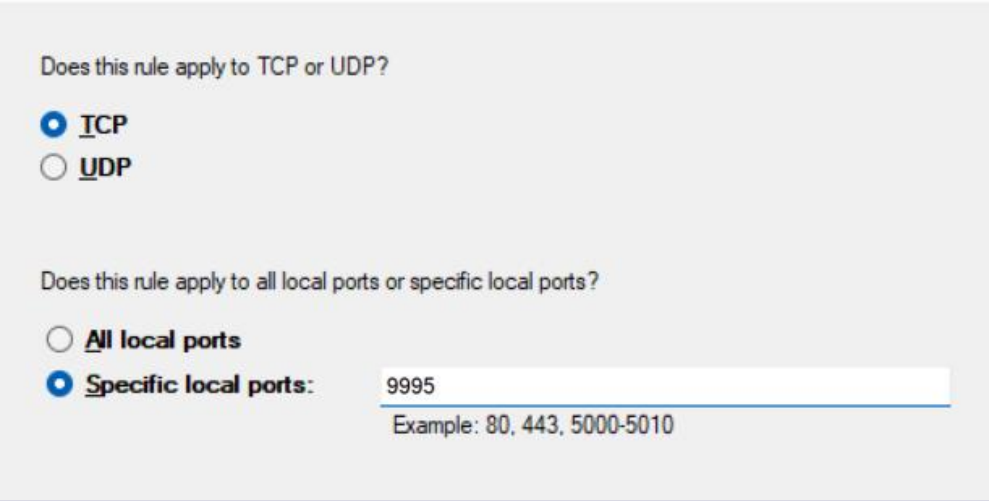




Select "Port" and click "Next."

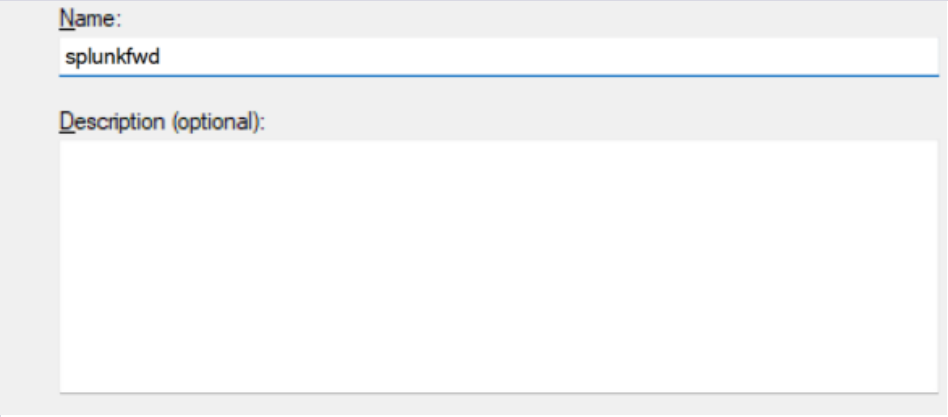


Choose “TCP” and enter the **<splunk_listener_port>** you assigned before(eg:9997) and click next.



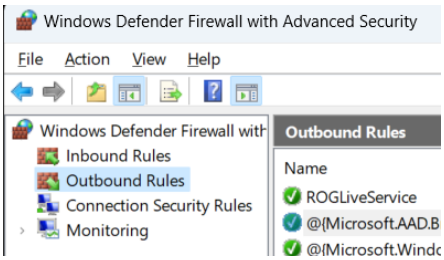
“Allow the connection” and click “Next”.

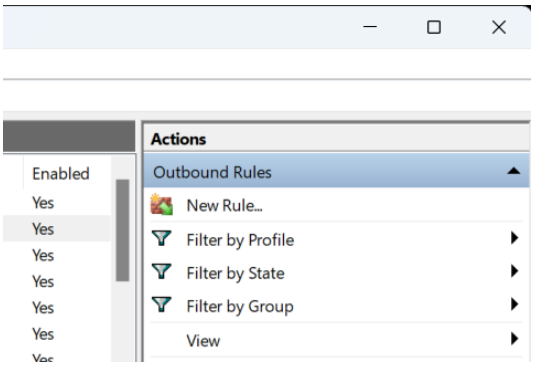
Enter a name for the rule (e.g., Splunk Universal Forwarder).



Click "Finish" to create the rule.

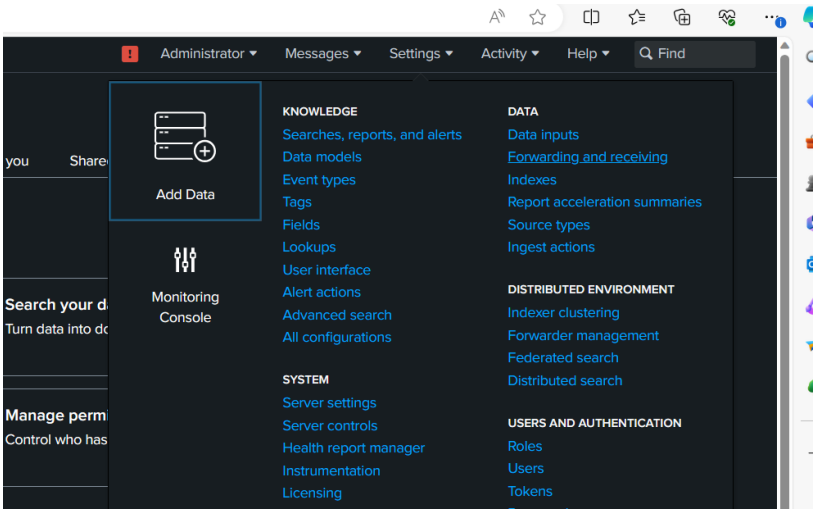
Same way set “Outbound Rules” also.



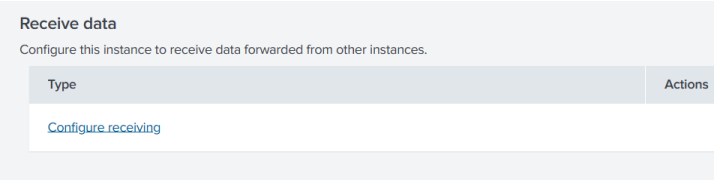


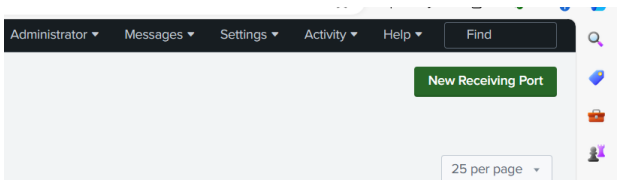
Splunk Enterprise configuration

Log in to your Splunk Enterprise. Go to "Settings" > "Forwarding and receiving".



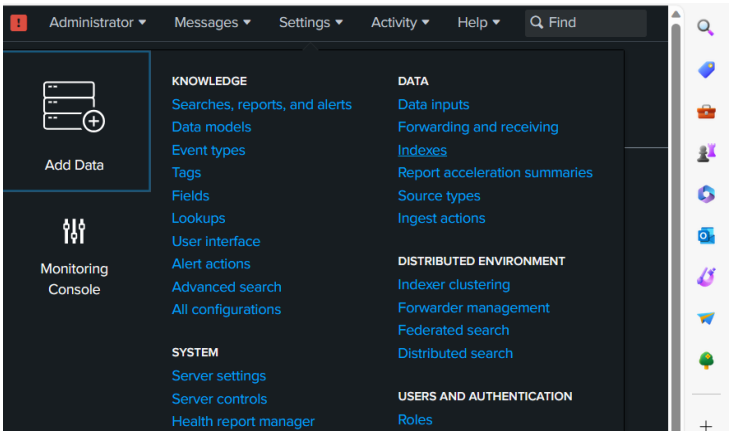
Click on “Configure receiving” and “New receiving port”.



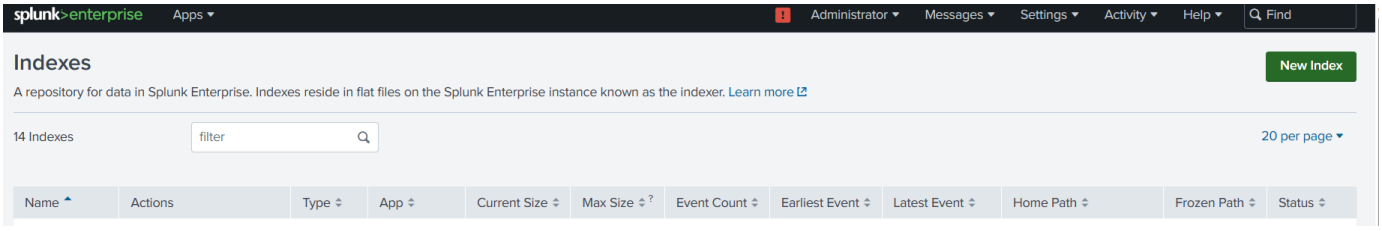


enter the <splunk_listener_port> you assigned before(eg:9997) and save.

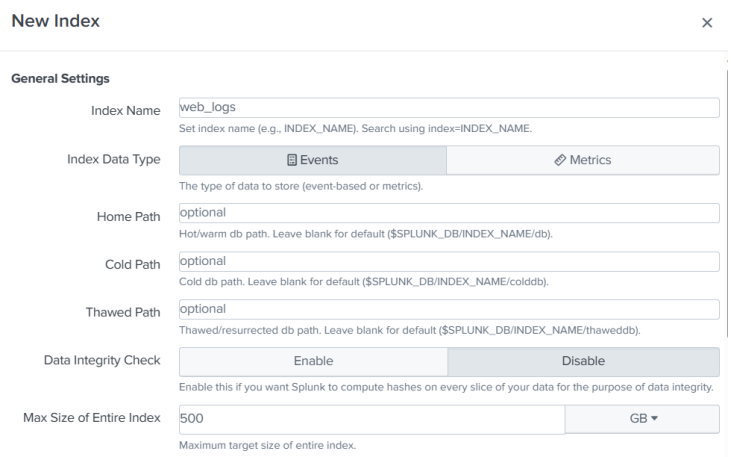
Then go to “Settings”>”Indexes”.



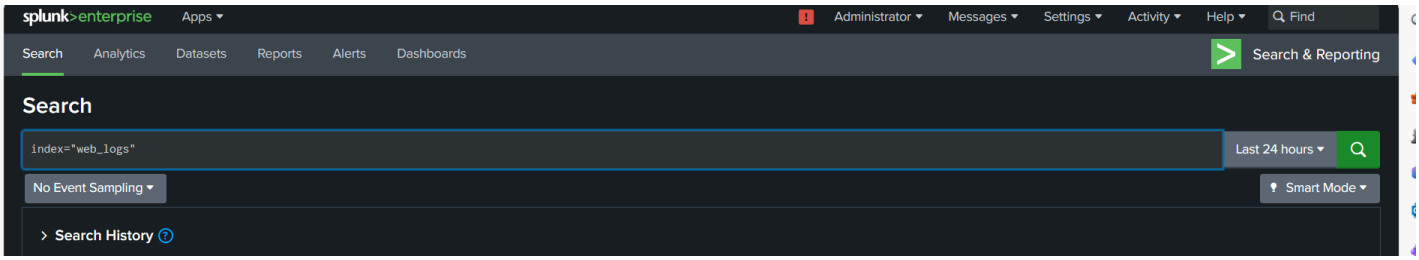
Ensure that the "web_logs" index is configured and enabled. If it's not, create the index and enable it. To create: click on “New Index”.



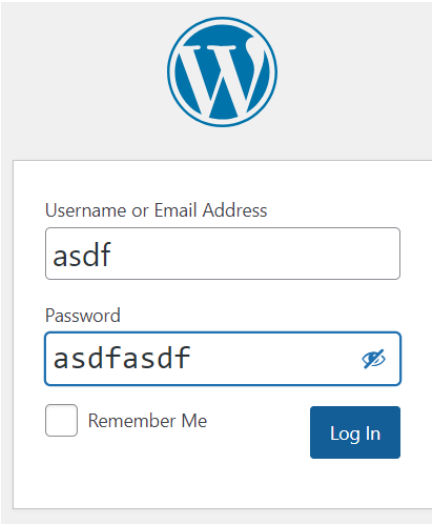
Give index name as “web_logs” and save it.

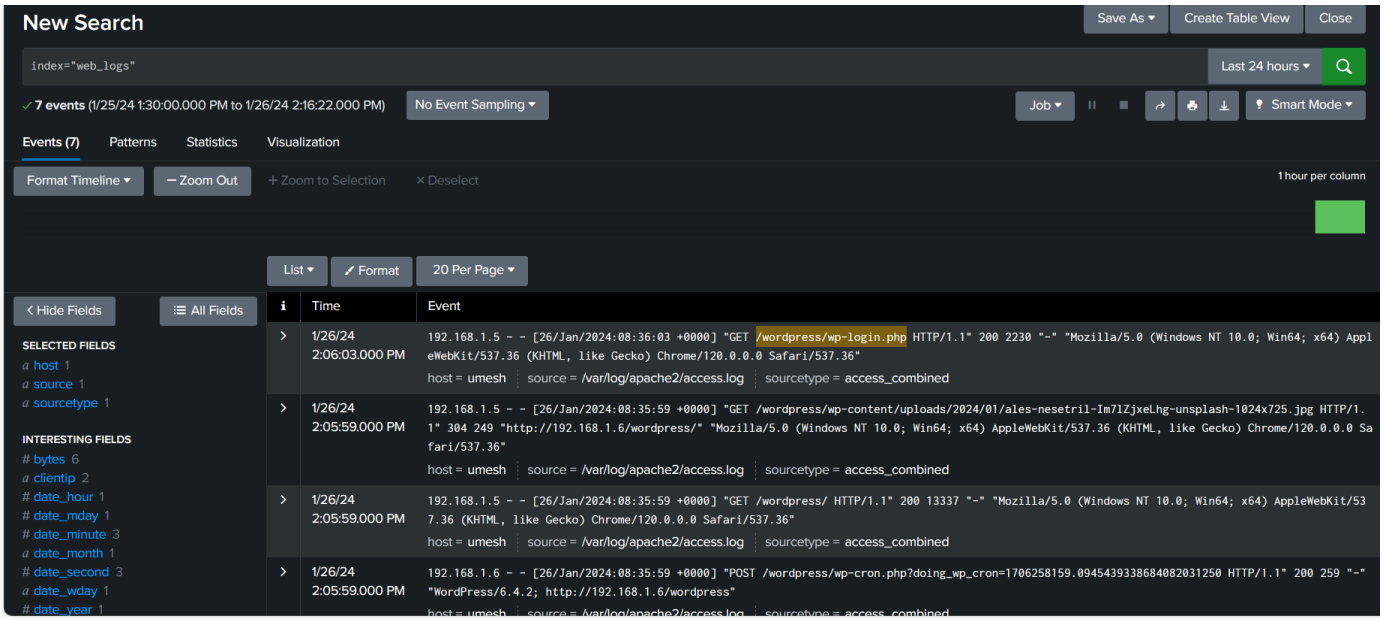


Then go to “Search & Reporting”. There search for: index=”web_logs”



Then go to WordPress site and perform some incorrect login events and come back to your Splunk enterprise refresh your search.





By following these steps, you should be able to monitor the WordPress site logs from your windows machine through Splunk enterprise,

Conclusion

Hopefully , this guide provides a step-by-step walkthrough for installing WordPress on Ubuntu 22.04, utilizing Apache as the hosting server, and implementing Splunk for effective log monitoring. By following these instructions, users can not only ensure a seamless setup of their WordPress site but also gain the tools and knowledge needed to monitor logs, track performance, troubleshoot issues, and maintain the health and smooth operation of their online presence. Whether you're a novice embarking on the journey of creating your first website or an experienced user seeking to enhance your online capabilities, this guide aims to simplify the processes of installation and monitoring, empowering you to establish and maintain a strong and resilient online presence.