Herramientas opcionales

Instalación apache y configuración de la página web

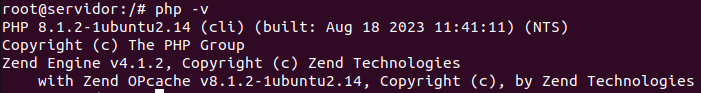
Lo primero, obviamente, es configurar apache. Si no lo tenemos instalado, se debe instalar.



Tras ello, instalamos PHP.

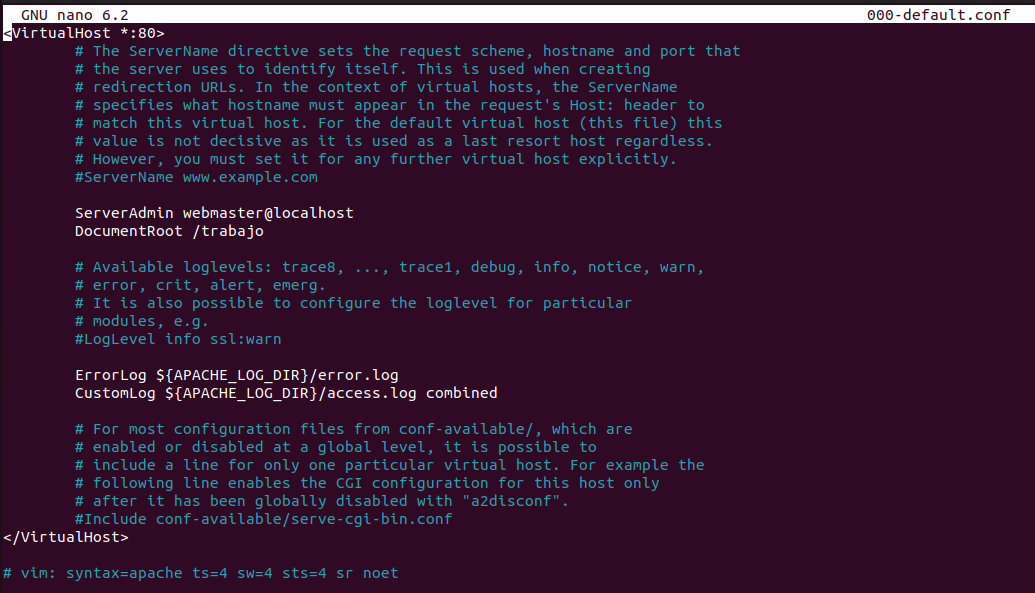


Revisamos la versión de PHP con el siguiente comando.

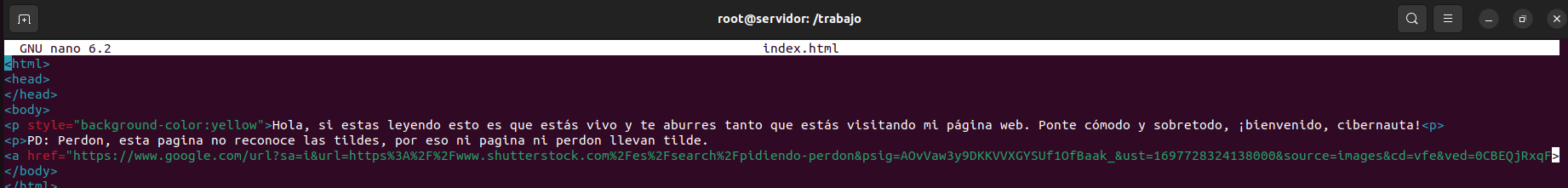


Ahora debemos ir al archivo 000-default.conf en /etc/apache2/sites-enabled para poder configurar el document root donde cambiaremos /var/www/html por el directorio donde vayamos a poner nuestra página HTML. En nuestro caso lo alojaremos en el directorio /trabajo.

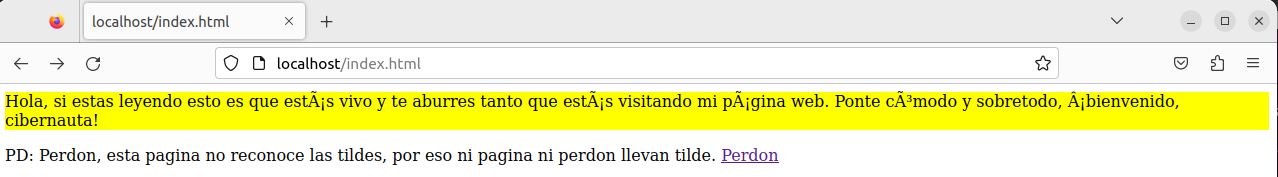




Dentro del directorio /trabajo crearemos un index.html que será el fichero donde configuremos nuestra página web. Lo editaremos con el comando nano y pondremos “Hola mundo” para ver si lo reconoce a buscarlo en el navegador.



En el buscador del navegador de nuestro ordenador podremos buscar la página a través de “localhost/”, “localhost/index.html” o en base a nuestra dirección IP local.



Una vez nos aparece, podremos configurar la página web a nuestro gusto.

Instalación PHP y configuración Xdebug

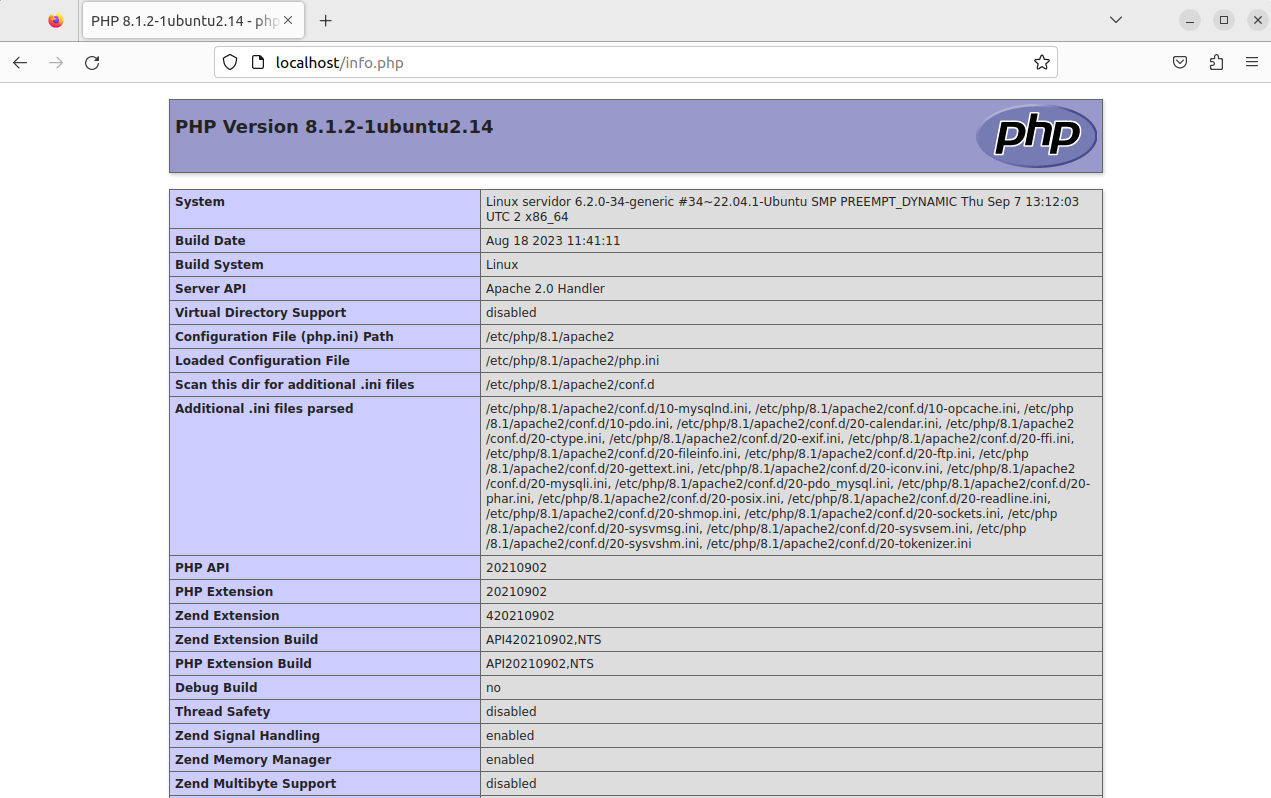
En el apartado interior ya hemos instalado PHP. Ahora en nuestro directorio de trabajo crearemos un documento php llamado info.php



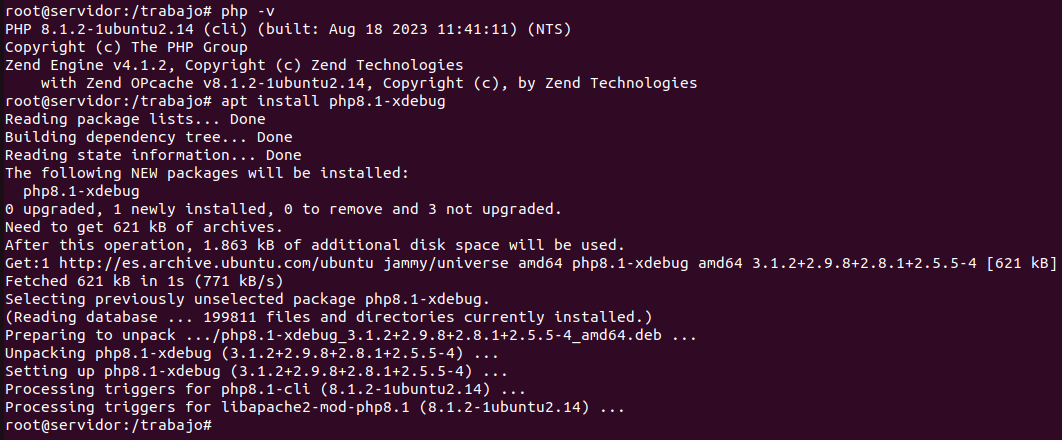
Dentro de éste configuraremos PHP.



Para saber si está bien configurado en el buscador ponemos localhost/info.php y nos deberá aparecer la página por defecto de php.



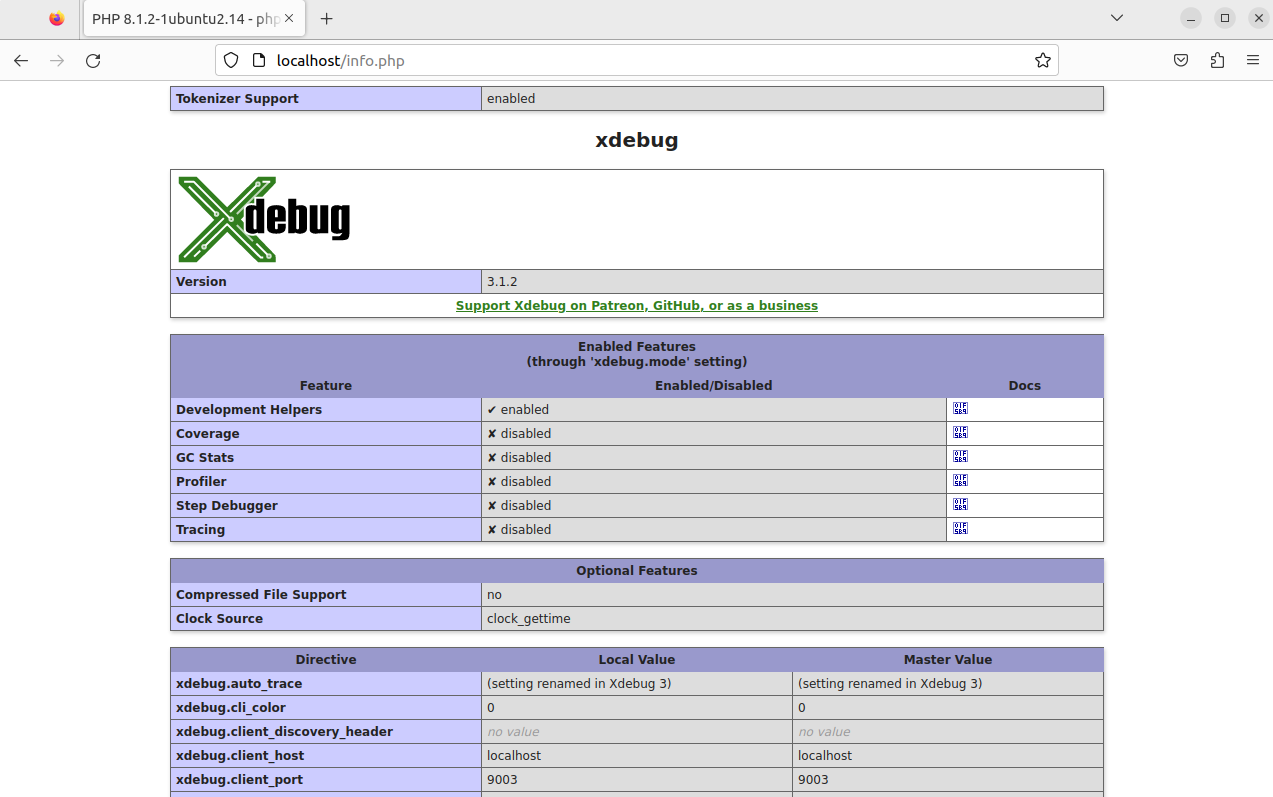
Para configurar el Xdebug deberíamos buscar la versión de PHP con php -v , como hemos visto anteriormente. En base a la versión instalaremos el xdebug con el siguiente comando:



Tras instalarlo, recargamos nuestro servicio apache.

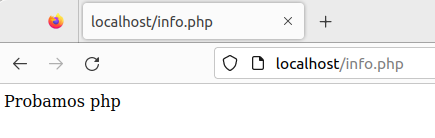


Para revisar si se ha instalado y configurado xdebug, vamos a nuestro buscador e introduciendo localhost/info.php podemos revisarlo bajando la barra.



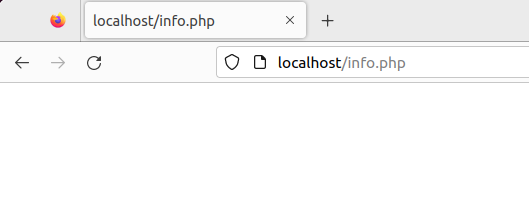
Configuramos el info.php para probar si funciona php.





Funciona perfectamente. Ahora meteremos texto al azar sin etiquetas.





Nos dará error ya que no aparecerá nada ya que habrá un error en el documento info.php. Recordemos que el xdebug sirve para que te marque, leyendo el documento de arriba hacia abajo, los diferentes fallos que puedan afectar al documento. Para que el xdebug actúe debemos configurarlo en la siguiente ruta:

/etc/php/8.1/apache2/php.ini

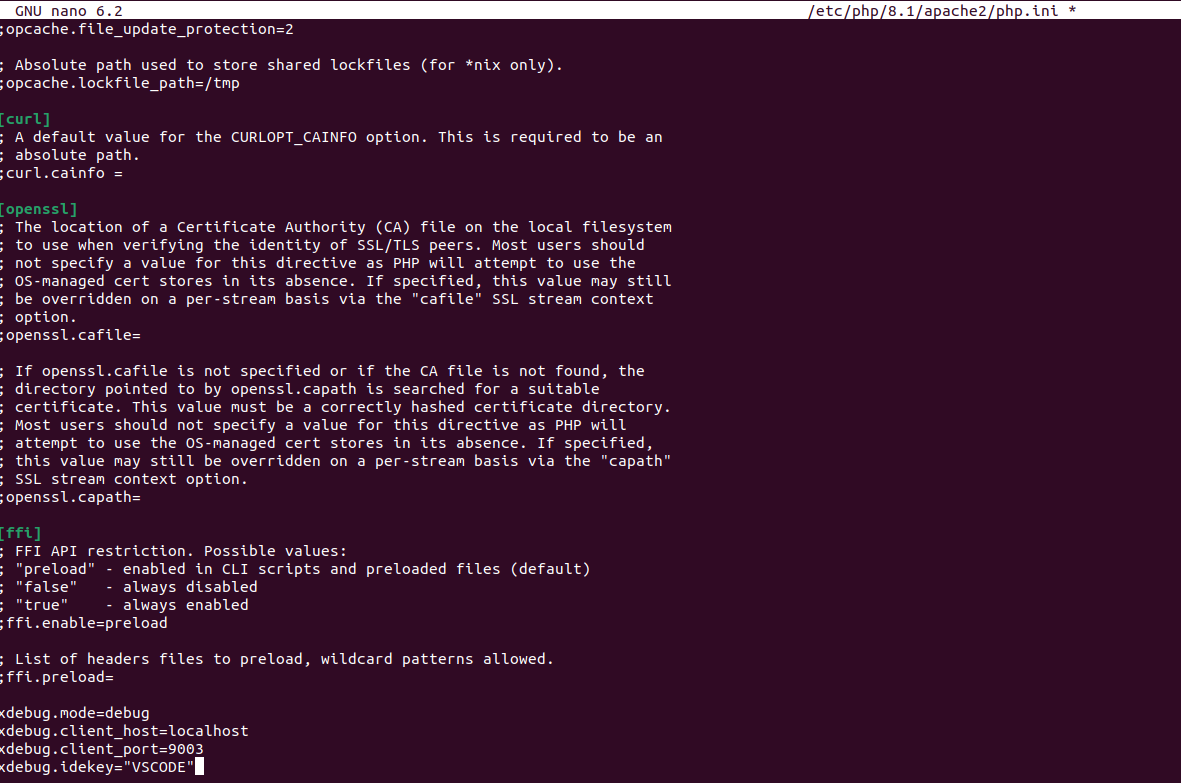
Dentro configuraremos el archivo php.ini como queramos nosotros configurarlo. Nos iremos al final del documento y escribiremos lo siguiente:

- xdebug.mode=debug (que trabaje en modo debug)

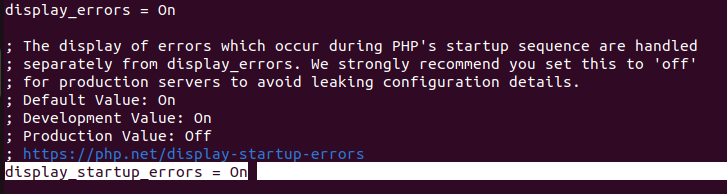
- xdebug.client\_host=localhost (que responda las peticiones localhost)

- xdebug.client\_port=9003 (por defecto que se conecte por el puerto)

- xdebug.idekey=”VSCODE” (le especficamos el ide que vamos a usar)



Hecho esto, pulsamos Ctrl + W y buscamos “display\_errors = Off”. Al hacerlo nos llevará a la línea y en ella deberemos poner On en lugar de Off, tanto en display\_errors como en display\_startup\_errors, que también estará en Off.



Tras guardar el documento con Ctrl + x, reiniciamos el servidor con systemctl restart apache2.

Al reiniciarlo e intentar acceder a localhost/php.ini, donde no nos aparecía nada, solamente la pantalla en blanco, ahora nos aparecerá el error en el documento php.



Nos dice que en la línea 3 de documento info.php dentro del directorio /trabajo hay un error de sintaxis, cosa que no aparecía anteriormente y nos ayudará en el futuro a poder corregir nuestros errores al desarrollar una página.

Instalación SSL

SSL es una herramienta o software que proporciona una certificación lets encrypt (abierta, automatizada y gratuita) que se instala en los servidores web para habilitar el protocolo HTTPS, protocolo principal usado para enviar datos entre un navegador web y un sitio web, aumentando así la seguridad de los datos.

Para hacerlo en nuestro servidor ejecutaremos el siguiente comando:



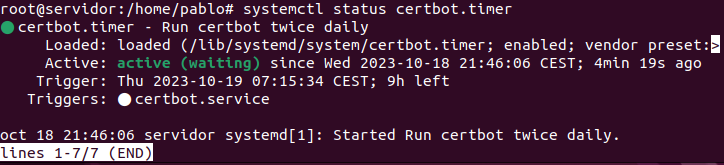
Como hemos estado trabajando con el servicio apache, para activarlo en nuestro servidor y dominio ejecutaremos el siguiente comando:



Nos pedirá el correo del dominio con el que lo hemos registrado.



Diremos yes, no, y luego introduciremos el nombre de nuestro dominio. Para ver si se ha activado o no, introduciremos el comando “systemctl status certbot.timer”. Al ver que está activo, reiniciaremos el servidor web, apache (systemctl restart apache2).



Instalación y uso de GIT

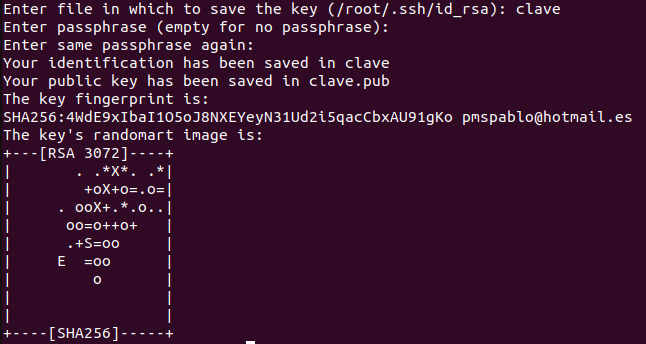
Lo primero que debemos hacer es instalar el software git. Como siempre, apt install git.

Una vez lo hayamos instalado deberemos darnos de alta en la web [www.github.com](http://www.github.com/).

Ya registrados debemos darnos de alta debemos crear una clave ssh para identificarnos:



Nos pedirá un fichero donde guardar la clave. En mi caso elegiré “clave”, para no complicarme y acordarme. También su contraseña.



Con cat clave.pub podemos ver la clave. Con ésta debemos irnos a [www.github.com](http://www.github.com/), y en la parte superior derecha, donde el icono de nuestro perfil, iremos a “Settings”. Dentro nos iremos a “SSH and GPG keys”, en el menú de la izquierda.



Le daremos a “new SSH key”, le pondremos un nombre, en mi caso asir.com y la clave que nos ha generado en el archivo clave.pub la pegaremos en el recuadro de abajo.



Podremos ahora clonar nuestro repositorio de GitHub. O bien con SSH, o bien con HTTPS. En mi caso lo haré con HTTPS. ¿Cómo? Con el siguiente comando.



La sintaxis de la ruta es comando git + parámetro clone + dirección https donde ponemos el dominio de github seguido por el nombre de usuario y por último el nombre de nuestro repositorio git.

En caso de haberlo hecho con SSH la sintaxis sería: git clone [git@github.com:pablotepalote/pablo.git](mailto:git@github.com:pablotepalote/pablo.git)

Una vez clonado, empezaremos a trabajar en el documento HTML de nuestra carpeta. Si no lo tenemos lo creamos.



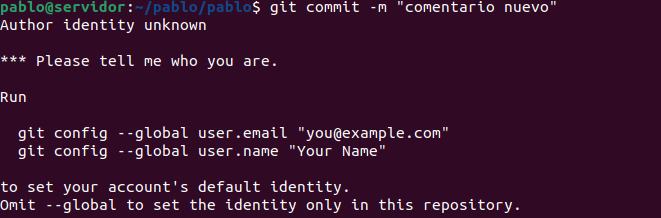
En mi caso ya lo tenía creado el documento hola.html, por lo que le he comentado nuevamente con un echo. Para poder subir nuestro archivo debemos ejecutar los comandos git add . o git add hola.html , el documento html creado previamente. Es preferible git add . debido a que añadiríamos todos los documentos al buffer, no sólo el archivo hola.html.



Debemos realizar ahora el comentario en la subida con el siguiente comando:

Git commit –m “nuevo comentario”

Al hacerlo nos saldrá el paso más importante para la subida a github: identificarnos.



Solamente debemos rellenarlo con el usuario de nuestra cuenta de github y en la siguiente orden con nuestro correo electrónico de github.



Una vez nos hayamos registrado podremos ya comentarlo con el comentario expuesto anteriormente.



Por último sólo quedaría el push del repositorio a la rama principal de nuestro repositorio Git con el comando “git push –u origin main” donde nos pedirá nuestros credenciales. Y listo.