

Índice

1.-Instalación y configuración del entorno de desarrollo.....	3
Documentación de la máquina CMHUSED:.....	3
-Creación de la máquina limpia:.....	3
+SSH:.....	4
-Instalación de software:.....	4
-Configuración del software:.....	4
-Mantenimiento:.....	4
-Monitorización:.....	4
-Copia de seguridad:.....	4
+Apache HTTP:.....	5
-Instalación de software:.....	5
-Configuración del software:.....	5
-Mantenimiento:.....	6
-Monitorización:.....	6
-Copia de seguridad:.....	6
+PHP:.....	6
-Instalación de software:.....	6
-Configuración del software:.....	6
-Mantenimiento:.....	7
-Monitorización:.....	7
-Copia de seguridad:.....	7
+MySQL:.....	7
-Instalación de software:.....	7
-Mantenimiento:.....	8
-Monitorización:.....	8
-Copia de seguridad:.....	8
Documentación de la máquina CMHWXED:.....	8
-Instalación de software:.....	8
-Prueba de software:.....	9
2.-Instalación y configuración del entorno de explotación personal.....	12
-Creación de la máquina limpia:.....	12
+SSH:.....	12
-Instalación de software:.....	12
-Configuración del software:.....	12
-Mantenimiento:.....	12
-Monitorización:.....	12
-Copia de seguridad:.....	13
+Apache HTTP:.....	13
-Instalación de software:.....	13
-Configuración del software:.....	13

-Mantenimiento:.....	14
-Monitorización:.....	14
-Copia de seguridad:.....	14
+PHP:.....	14
-Instalación de software:.....	14
-Configuración del software:.....	15
-Mantenimiento:.....	15
-Monitorización:.....	15
-Copia de seguridad:.....	15
+MySQL:.....	15
-Instalación de software:.....	15
-Configuración del software:.....	15
-Mantenimiento:.....	16
-Monitorización:.....	16
-Copia de seguridad:.....	16
Transferir archivos de un servidor a otro:.....	16

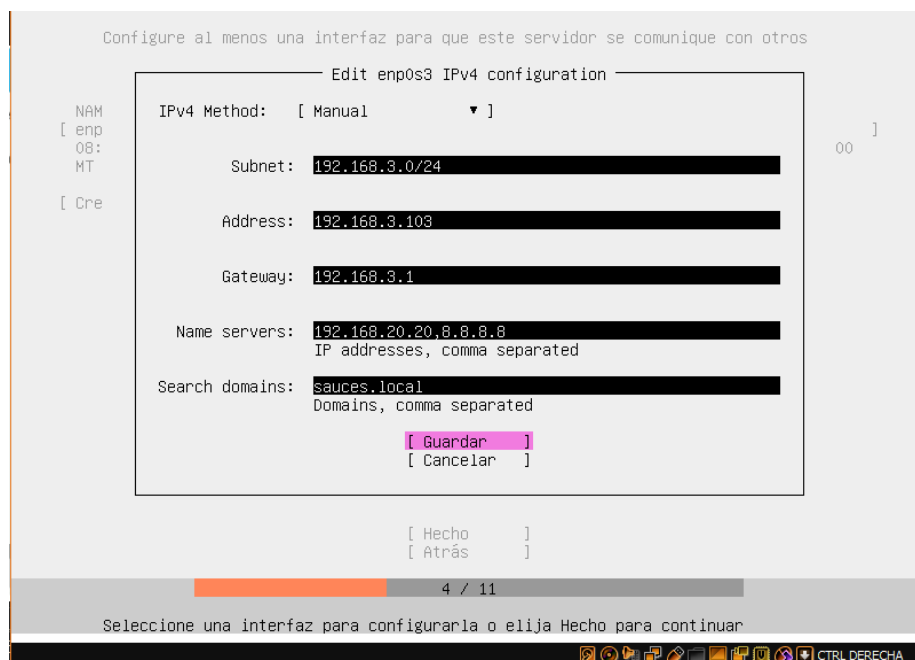
1.-Instalación y configuración del entorno de desarrollo.

Para la asignatura de Desarrollo web del lado servidor sobre dos máquinas virtuales xxxUSED y xxxW7ED. Documentando la configuración elegida y los aspectos mas importantes del proceso de instalación. Documentar el proceso de monitorización del correcto funcionamiento de los servicios. Mantener actualizada esta documentación durante todo el curso.

Documentación de la máquina CMHUSED:

-Creación de la máquina limpia:

Primero configuramos las opciones de idioma y demás, luego configuramos la red añadiendo +100 a la IP del anfitrión tal que así:



Posteriormente, configuramos las particiones de memoria, de los 500Gb asignados a la unidad 100 irán al sistema en el directorio, el doble de la RAM, en mi caso 8Gb, irán destinados al directorio swap, y el resto irán para datos. La cuenta principal que crearemos será miadmin/paso.

Para actualizar la lista de repositorios de Ubuntu descargamos un archivo `sources.list` actualizado, después debemos ir al directorio `/etc/apt/` y cambiar el nombre del archivo `sources.list` original por `sources.list.back` con el comando `mv sources.list sources.list.back`. Después transferimos el archivo nuevo con el Filezilla al home del usuario `/home/miadmin` y desde ese directorio lo movemos a su carpeta con el comando `mv sources.list /etc/apt/`. Ahora sólo queda ejecutar los siguientes comandos para actualizar los repositorios:

sudo apt-get update, sudo apt-get upgrade, sudo apt-get autoclean y sudo apt-get autoremove.

Para ver las particiones del disco usamos el comando lsblk -fm/df -h.

Si hemos clonado desde una máquina limpia, lo primero será cambiar el nombre de la máquina, para ello, modificamos el fichero hostname con el comando sudo nano /etc/hostname y poniendo el nombre a la máquina, en mi caso CMHUSED, luego ejecutamos el comando sudo hostnamectl set-hostname CMHUSED y para hacer permanente el cambio ejecutamos el comando sudo nano /etc/cloud/cloud.cfg y donde pone "preserve_hostname"=false, lo cambiamos a true, y ya tendremos el nombre de la máquina modificado, reiniciamos el servidor con el comando reboot y comprobamos con el comando hostname que el nombre se ha cambiado correctamente.

+SSH:

-Instalación de software:

El servicio SSH ya viene instalado por defecto en el Ubuntu 18.04, por lo tanto, no será necesaria su instalación.

-Configuración del software:

SSH ya viene configurado por defecto para su uso inmediatamente, así que no necesitamos modificar nada en principio. Los ficheros de configuración de SSH se encuentran en dos direcciones, el del **cliente** en /etc/ssh/ssh_config y el del **servidor** en /etc/ssh/sshd_config.

-Mantenimiento:

Podemos parar, iniciar o reiniciar el servicio con el comando sudo service ssh stop/start/reboot/restart respectivamente.

-Monitorización:

Podemos comprobar el estado del servicio SSH con el comando sudo service ssh status.

-Copia de seguridad:

Para guardar la configuración de SSH en caso de querer cargarla en otra máquina o por razones de seguridad deberemos copiar el archivo de configuración /etc/ssh/sshd_config en nuestra máquina anfitriona.

+Apache HTTP:

-Instalación de software:

Para comenzar a instalar el servidor Apache usaremos el comando **sudo apt-get install apache2.**

-Configuración del software:

Si nos vamos al fichero de configuración de puertos con **sudo nano /etc/apache2/ports.conf** podemos cambiar el puerto por el que se comunica Apache o incluso añadir mas puertos de la siguiente manera:

```
Listen 80
Listen 81
Listen 82
<IfModule ssl_module>
    Listen 443
</IfModule>

<IfModule mod_gnutls.c>
    Listen 443
</IfModule>
```

A continuación para poder subir archivos desde el anfitrión con FileZilla, NetBeans, Notepad++ u otros programas deberemos cambiar los permisos en la carpeta que sirve Apache desde el servidor, para esto usamos el comando **sudo chmod 777 /var/www/html** para hacerlo de forma rápida aunque no es una forma segura. También podemos crear un usuario seguro que subirá los archivos como operadorweb perteneciendo al grupo *www-data*, para crear el usuario usamos el comando:

sudo adduser --ingroup www-data --home /var/www/html --no-create-home operadorweb.

Para comprobar que está creado el usuario usamos el comando **cat /etc/passwd | grep operadorweb** y para hacerlo propietario de la carpeta para poder subir archivos usamos el comando:

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

Luego cambiamos los permisos de la carpeta html con el comando **sudo chmod -R 775 /var/www/html.** Para ver el propietario de la carpeta usamos el comando **ls -l /var/www.** Una vez hecho esto, ya podremos subir archivos al servidor desde el cliente.

Para ver los ficheros de Apache podemos usar el comando **apache2ctl -S.**

Para ver los módulos de Apache se usa el comando **/usr/sbin/apache2ctl -t -D DUMP_MODULES.** Para activar un módulo de Apache nos vamos al directorio *mods-enabled* con el comando **cd /etc/apache2/mods-enabled** y ejecutamos el comando **sudo a2enmod "nombre_módulo"** o con **a2dismod** en caso de querer desactivarlo. Para ver los módulos activos usamos el comando **apache2ctl -M.**

Para cambiar la extensión del índice que se abre de forma predeterminada en nuestro servidor podemos editarlo con **sudo nano /etc/apache2/mods-enabled/dir.conf** y podremos cambiar el orden de preferencia de extensiones.

-Mantenimiento:

En Apache, para reiniciar el servicio debemos ejecutar el comando **sudo service apache2 restart/reload**, también podemos pararlo o iniciarlo con **stop/start**, para cambiar la carpeta desde la que sirve Apache lo podemos modificar con **sudo nano /etc/apache2/sites-enabled/000-default.conf** y modificando en DocumentRoot la ruta predeterminada de Apache **/var/www/html** por la que queramos nosotros.

-Monitorización:

Podemos ver el estado de Apache con el comando **service apache2 status**:

```
miadmin@CMHUSED:~$ service apache2 status
• apache2.service - The Apache HTTP Server
  Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
  Drop-In: /lib/systemd/system/apache2.service.d
           └─apache2-systemd.conf
  Active: active (running) since Fri 2018-09-21 09:49:12 UTC; 12min ago
```

Podemos ver los puertos que están abiertos con el comando **netstat -ltn**. También podemos ver la lista de app instaladas con **sudo ufw app list**.

-Copia de seguridad:

Si queremos quedarnos con la configuración de Apache debemos guardar en el anfitrión los archivos ubicados en **/etc/apache2/sites-enabled/000-default.conf** y **/etc/apache2/ports.conf**.

+PHP:

-Instalación de software:

Instalaremos el módulo de PHP con el comando **sudo apt-get install php** que nos instalará automáticamente la última versión disponible, a continuación instalaremos el módulo de PHP para usar MySQLi con el comando **sudo apt-get install php-mysqli** y el módulo para trabajar con ficheros xml con el comando **sudo apt-get install php-xml**.

-Configuración del software:

El fichero de configuración de PHP se puede modificar con el comando **sudo nano /etc/php/7.2/apache2/php.ini**. Es importante configurar el parámetro **upload_max_filesize** con un valor no muy elevado para evitar que puedan tirar nuestro servidor colapsando su capacidad de almacenamiento mediante la subida de archivos muy

grandes. Además, el parámetro `error_reporting`, que indica el tipo de errores que se mostrarán, debe ser distinto dependiendo de si lo empleamos en nuestro servidor de desarrollo o explotación. También podemos configurar el parámetro `display_errors` poniéndolo en "On" para que nos muestre los errores de PHP que pueda haber en nuestros programas. Los módulos predeterminados de php se encuentran en el directorio `/usr/lib/apache2/modules/libphp7.2.so`.

```
; display_errors
;   Default Value: On
;   Development Value: On
;   Production Value: Off

; display_startup_errors
;   Default Value: Off
;   Development Value: On
;   Production Value: Off

; error_reporting
;   Default Value: E_ALL & ~E_NOTICE & ~E_STRICT & ~E_DEPRECATED
;   Development Value: E_ALL
;   Production Value: E_ALL & ~E_DEPRECATED & ~E_STRICT
```

-Mantenimiento:

Puesto que PHP no es un servicio, no podemos pararlo ni iniciarlo, ni reiniciarlo, y en principio no requiere de mantenimiento.

-Monitorización:

Para comprobar que se ha instalado el PHP correctamente lo veremos con el comando `php -v`, para ver los módulos que tiene PHP se usa el comando `apt-cache search php`.

-Copia de seguridad:

Si queremos guardar la configuración de PHP debemos hacer una copia del archivo ubicado en el path `/etc/php/7.2/apache2/php.ini`.

+MySQL:

-Instalación de software:

Para instalar la base de datos MySQL usamos el comando `sudo apt-get install mysql-server`. También debemos instalar dos módulos de PHP para MySQL con el comando `sudo apt-get install -y php-mysql php-cli`.

-Configuración del software:

Para poner una contraseña en el root de MySQL entramos desde consola con `mysql -u root -p`. Entramos a la base de datos mysql con el comando `use mysql;` y ejecutamos los siguientes comandos

SQL:

```
flush privileges;  
update user set authentication_string=PASSWORD("contraseña")  
where user='root';  
update user set plugin="mysql_native_password" where  
user='root';
```

Luego salimos de mysql con el comando exit. Y por ultimo reiniciamos el servicio con sudo service mysql restart.

También puede hacerse con un sólo comando: alter user 'root'@'localhost' identified with mysql_native_password by 'paso';

Los ficheros de configuración de MySQL podremos configurarlos con el comando sudo nano /etc/mysql/my.cnf y con sudo nano /etc/mysql/mysql.cnf y desde ellos podremos configurar varios comandos como por ejemplo para la gestión de usuarios y también en la ruta /etc/mysql/mysql.conf.d/mysqld.cnf tenemos el archivo de configuración principal donde podemos cambiar el puerto por el que actúa, conexiones máximas...

-Mantenimiento:

Podemos reiniciar, parar e iniciar el servicio MySQL con el comando sudo service MySQL reload/restart/stop/start, respectivamente.

-Monitorización:

En MySQL podemos ver el estado del servicio con el comando sudo service mysql status.

-Copia de seguridad:

Si queremos guardar la configuración de MySQL debemos guardar los archivos ubicados en /etc/mysql/my.cnf, /etc/mysql/mysql.cnf y /etc/mysql/mysql.conf.d/mysqld.cnf.

Documentación de la máquina CMHWXED:

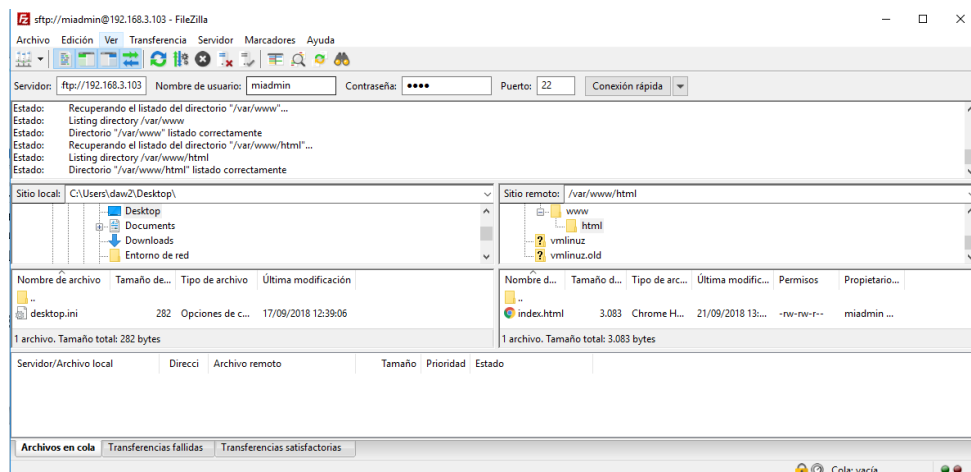
-Instalación de software:

Para poder comunicarnos y/o subir archivos al servidor necesitaremos algún programa como NetBeans, Fillezilla, Putty... que son gratuitos y los podremos descargar de sus páginas oficiales e instalarlos siguiendo los pasos del instalador. Para probar MySQL descargaremos e instalaremos el WorkBench de la página oficial.

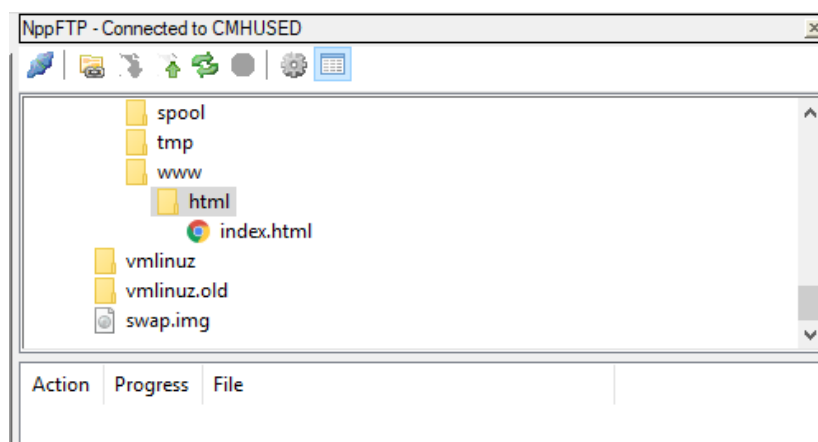
-Prueba de software:

Para probar el Apache, con el servidor arrancado, en el navegador de wx de nuestra máquina cliente entramos a la IP de nuestro servidor, en mi caso 192.168.3.103, y debería aparecer el index que hemos subido previamente mediante una aplicación con SFTP, como se muestra a continuación en las siguientes capturas:

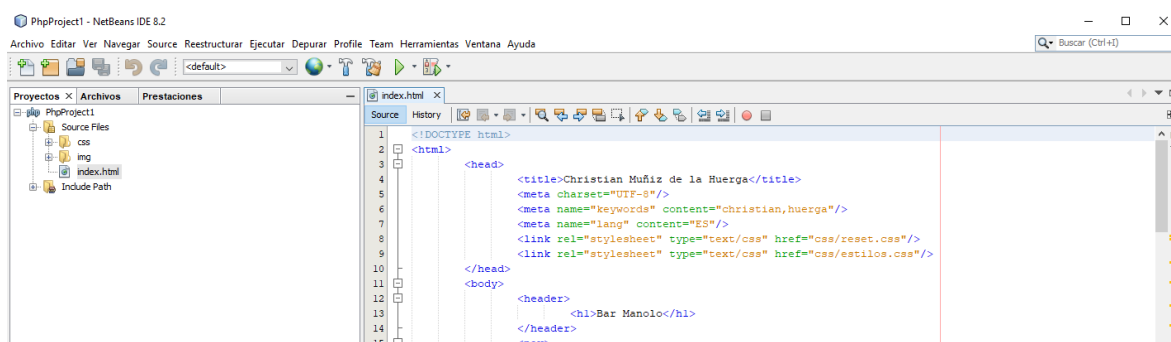
Filezilla:



NotePad++:

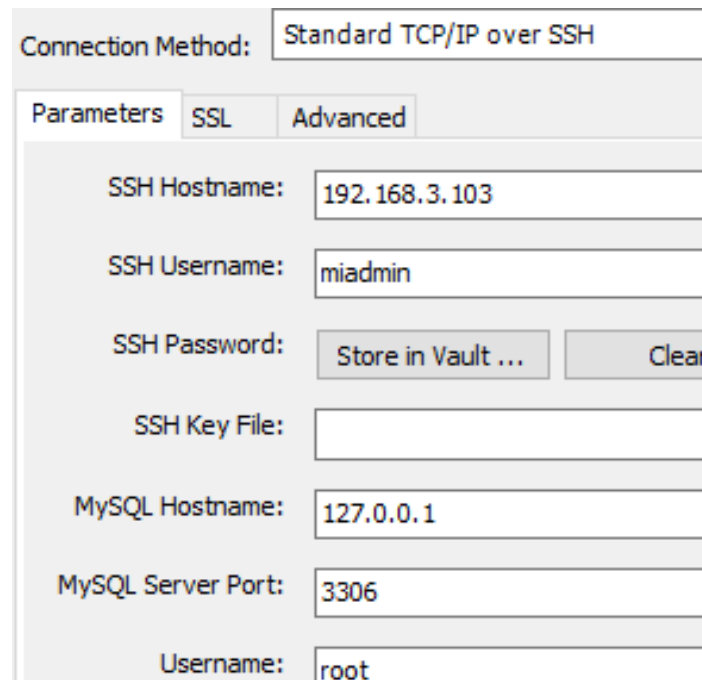


NetBeans:



Para probar el PHP insertamos alguna instrucción PHP en nuestro índice para comprobar que funciona.

Ahora para conectarnos desde el anfitrión con el workBench creamos una nueva conexión con los datos de la siguiente manera:

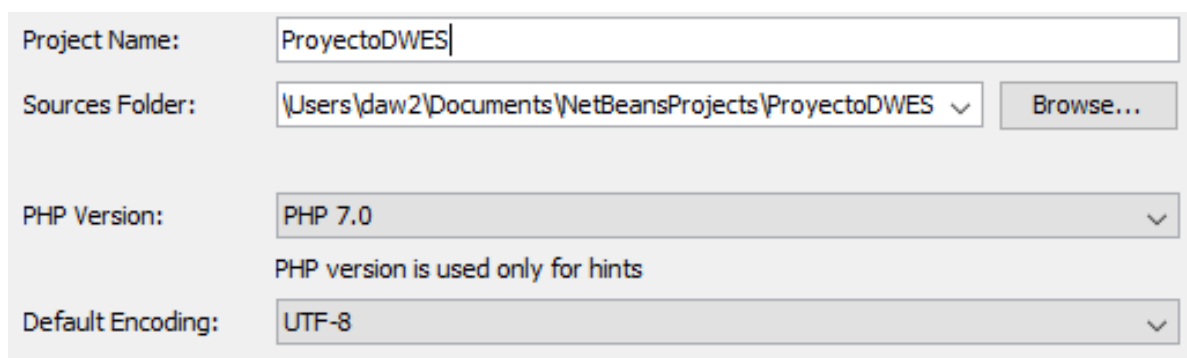


The screenshot shows the 'Standard TCP/IP over SSH' connection configuration window in NetBeans. It has three tabs: 'Parameters', 'SSL', and 'Advanced'. The 'Parameters' tab is active, showing the following fields:

- SSH Hostname: 192.168.3.103
- SSH Username: miadmin
- SSH Password: (with buttons for 'Store in Vault ...' and 'Clear')
- SSH Key File: (empty field)
- MySQL Hostname: 127.0.0.1
- MySQL Server Port: 3306
- Username: root

Para ejecutar un script sql en workBench nos vamos a File, Run SQL Script y abrimos el archivo con el script.

Para crear un proyecto en NetBeans en su servidor remoto le damos dentro de NetBeans a nuevo proyecto de PHP, PHP Application from Remote Server:



The screenshot shows the 'New Project' dialog in NetBeans. The 'Project Name' is 'ProyectoDWES'. The 'Sources Folder' is set to a local path, with a 'Browse...' button next to it. The 'PHP Version' is set to 'PHP 7.0'. Below this, it says 'PHP version is used only for hints'. The 'Default Encoding' is set to 'UTF-8'.

Después de poner el nombre del proyecto y versión de PHP, en Project URL ponemos la IP de nuestro servidor, y le damos a Manage para configurar los datos del servidor:

Name: CMHUSED

Host Name: 192.168.3.103 Port: 22

User Name: miadmin

Password: ••••

Leave password empty to be prompted.
(or specify Private Key file)

Private Key File: Browse...

Running ssh-agent will be used.

Known Hosts File: Browse...

Initial Directory: /var/www/html

Timeout (s): 30

Keep-alive interval (s): 30 Interval 0 means disabled.

Le damos a siguiente y nos aparecerán los archivos que ya existen en el servidor para poder descargarlos en el proyecto.

Confirmation

The following files will be downloaded after you click Finish.
The download process cannot be cancelled once started!

- SourceFiles
 - ☒ PhpProject1
 - ☒ ProyectoDWES
 - ☒ index.html

☒ Check All

3 files (or more) selected i Selected files will be downloaded, existing files will be overwritten.

Project files will be automatically uploaded to the server every time you edit any of them.

< Atrás Siguiente > Terminar Cancelar Ayuda

Le damos a terminar y ya tendremos nuestro proyecto creado.

2.-Instalación y configuración del entorno de explotación personal.

Para la asignatura de Desarrollo web del lado servidor sobre una máquina virtual xxx-USEE. Documentando la configuración elegida y los aspectos mas importantes del proceso de instalación. Documentar el proceso de monitorización del correcto funcionamiento de los servicios. Creación y actualización del ProyectoDWES durante el curso.

-Creación de la máquina limpia:

Clonaremos nuestra máquina limpia CMHUSLimpia, y tendremos que cambiar el nombre de la máquina, para ello, modificamos el fichero `hostname` con el comando `sudo nano /etc/hostname` y poniendo el nombre a la máquina, en mi caso CMHUSEE, luego ejecutamos el comando `sudo hostnamectl set-hostname CMHUSEE` y para hacer permanente el cambio ejecutamos el comando `sudo nano /etc/cloud/cloud.cfg` y donde pone "preserve_hostname"=false, lo cambiamos a true, una vez hecho deberemos cambiar la configuración de red añadiendo +200 a la IP del anfitrión, para ello usamos el comando `sudo nano /etc/netplan/50-cloud-init.yaml` y en addresses añadimos 200 a la IP del servidor, en mi caso quedaría 192.168.2.203, luego ponemos el comando `sudo netplan apply` y reiniciamos la máquina con el comando `reboot`.

+SSH:

-Instalación de software:

El servicio SSH ya viene instalado por defecto en el Ubuntu 18.04, por lo tanto, no será necesaria su instalación.

-Configuración del software:

SSH ya viene configurado por defecto para su uso inmediatamente, así que no necesitamos modificar nada en principio. Los ficheros de configuración de SSH se encuentran en dos direcciones, el del cliente en `/etc/ssh/ssh_config` y el del servidor en `/etc/ssh/sshd_config`.

-Mantenimiento:

Podemos parar, iniciar o reiniciar el servicio con el comando `sudo service ssh stop/start/reboot/restart` respectivamente.

-Monitorización:

Podemos comprobar el estado del servicio SSH con el comando `sudo service ssh status`.

-Copia de seguridad:

Para guardar la configuración de SSH en caso de querer cargarla en otra máquina o por razones de seguridad deberemos copiar el archivo de configuración `/etc/ssh/sshd_config` en nuestra máquina anfitriona.

+Apache HTTP:

-Instalación de software:

Para comenzar a instalar el servidor Apache usaremos el comando `sudo apt-get install apache2`.

-Configuración del software:

Si nos vamos al fichero de configuración de puertos con `sudo nano /etc/apache2/ports.conf` podemos cambiar el puerto por el que se comunica Apache o incluso añadir mas puertos de la siguiente manera:

```
Listen 80
Listen 81
Listen 82
<IfModule ssl_module>
    Listen 443
</IfModule>

<IfModule mod_gnutls.c>
    Listen 443
</IfModule>
```

A continuación para poder subir archivos desde el anfitrión con FileZilla, NetBeans, Notepad++ u otros programas deberemos cambiar los permisos en la carpeta que sirve Apache desde el servidor, para esto usamos el comando `sudo chmod 777 /var/www/html` para hacerlo de forma rápida aunque no es una forma segura. También podemos crear un usuario seguro que subirá los archivos como operadorweb perteneciendo al grupo `www-data`, para crear el usuario usamos el comando:

`sudo adduser --ingroup www-data --home /var/www/html --no-create-home operadorweb`.

Para comprobar que está creado el usuario usamos el comando `cat /etc/passwd | grep operadorweb` y para hacerlo propietario de la carpeta para poder subir archivos usamos el comando:

`sudo chown -R operadorweb:www-data /var/www/html`

Luego cambiamos los permisos de la carpeta `html` con el comando `sudo chmod -R 775 /var/www/html`. Para ver el propietario de la carpeta usamos el comando `ls -l /var/www`. Una vez hecho esto, ya podremos subir archivos al servidor desde el cliente.

Para ver los ficheros de Apache podemos usar el comando `apache2ctl`

-S.

Para ver los módulos de Apache se usa el comando /usr/sbin/apache2ctl -t -D DUMP_MODULES. Para activar un módulo de Apache nos vamos al directorio *mods-enabled* con el comando cd /etc/apache2/mods-enabled y ejecutamos el comando sudo a2enmod "nombre_módulo" o con a2dismod en caso de querer desactivarlo. Para ver los módulos activos usamos el comando apache2ctl -M.

Para cambiar la extensión del índice que se abre de forma predeterminada en nuestro servidor podemos editarlo con sudo nano /etc/apache2/mods-enabled/dir.conf y podremos cambiar el orden de preferencia de extensiones.

-Mantenimiento:

En Apache, para reiniciar el servicio debemos ejecutar el comando sudo service apache2 restart/reload, también podemos pararlo o iniciarlo con stop/start, para cambiar la carpeta desde la que sirve Apache lo podemos modificar con sudo nano /etc/apache2/sites-enabled/000-default.conf y modificando en DocumentRoot la ruta predeterminada de Apache /var/www/html por la que queramos nosotros.

-Monitorización:

Podemos ver el estado de Apache con el comando service apache2 status:

```
miadmin@CMHUSED:~$ service apache2 status
• apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Drop-In: /lib/systemd/system/apache2.service.d
             └─apache2-systemd.conf
   Active: active (running) since Fri 2018-09-21 09:49:12 UTC; 12min ago
```

Podemos ver los puertos que están abiertos con el comando netstat -ltn. También podemos ver la lista de app instaladas con sudo ufw app list.

-Copia de seguridad:

Si queremos quedarnos con la configuración de Apache debemos guardar en el anfitrión los archivos ubicados en /etc/apache2/sites-enabled/000-default.conf y /etc/apache2/ports.conf.

+PHP:

-Instalación de software:

Instalaremos el módulo de PHP con el comando sudo apt-get install php que nos instalará automáticamente la última versión disponible, a continuación instalaremos el módulo de PHP para usar MySQLi con el comando sudo apt-get install php-mysqli y el módulo para trabajar con ficheros xml con el comando sudo apt-get install php-xml.

-Configuración del software:

El fichero de configuración de PHP se puede modificar con el comando sudo nano /etc/php/7.2/apache2/php.ini. Es importante configurar el parámetro upload_max_filesize con un valor no muy elevado para evitar que puedan tirar nuestro servidor colapsando su capacidad de almacenamiento mediante la subida de archivos muy grandes. Además, el parámetro error_reporting, que indica el tipo de errores que se mostrarán, debe ser distinto dependiendo de si lo empleamos en nuestro servidor de desarrollo o explotación. Los módulos predeterminados de php se encuentran en el directorio /usr/lib/apache2/modules/libphp7.2.so.

-Mantenimiento:

Puesto que PHP no es un servicio, no podemos pararlo ni iniciarlo, ni reiniciarlo, y en principio no requiere de mantenimiento.

-Monitorización:

Para comprobar que se ha instalado el PHP correctamente lo veremos con el comando php -v, para ver los módulos que tiene PHP se usa el comando apt-cache search php.

-Copia de seguridad:

Si queremos guardar la configuración de PHP debemos hacer una copia del archivo ubicado en el path /etc/php/7.2/apache2/php.ini.

+MySQL:

-Instalación de software:

Para instalar la base de datos MySQL usamos el comando sudo apt-get install mysql-server. También debemos instalar dos módulos de PHP para MySQL con el comando sudo apt-get install -y php-mysql php-cli.

-Configuración del software:

Para poner una contraseña en el root de MySQL entramos desde consola con mysql -u root -p. Entramos a la base de datos mysql con el comando use mysql; y ejecutamos los siguientes comandos SQL:

flush privileges;

update user set authentication_string=PASSWORD("contraseña") where user='root';

update user set plugin="mysql_native_password" where user='root';

Luego salimos de mysql con el comando exit. Y por ultimo reiniciamos el servicio con sudo service mysql restart.

También puede hacerse con un sólo comando: alter user 'root'@'localhost' identified with mysql_native_password by 'paso';

Los ficheros de configuración de MySQL podremos configurarlos con el comando sudo nano /etc/mysql/my.cnf y con sudo nano /etc/mysql/mysql.cnf y desde ellos podremos configurar varios comandos como por ejemplo para la gestión de usuarios y también en la ruta /etc/mysql/mysql.conf.d/mysqld.cnf tenemos el archivo de configuración principal donde podemos cambiar el puerto por el que actúa, conexiones máximas...

-Mantenimiento:

Podemos reiniciar, parar e iniciar el servicio MySQL con el comando sudo service MySQL reload/restart/stop/start, respectivamente.

-Monitorización:

En MySQL podemos ver el estado del servicio con el comando sudo service mysql status.

-Copia de seguridad:

Si queremos guardar la configuración de MySQL debemos guardar los archivos ubicados en /etc/mysql/my.cnf, /etc/mysql/mysql.cnf y /etc/mysql/mysql.conf.d/mysqld.cnf.

Transferir archivos de un servidor a otro:

Nos conectaremos con el Putty al servidor que queremos transferir, en este caso, al de explotación, nos vamos a la carpeta sobre la que actúa el Apache, /var/www/html y desde allí nos conectamos por sftp a la máquina de desarrollo con el comando sftp miadmin@192.168.3.103, siendo miadmin el nombre de usuario y 192.168.3.103 la IP en la máquina de desarrollo, nos pedirá la contraseña de usuario, la introducimos y ya estaremos dentro de la máquina. Nos vamos a la carpeta donde sirve Apache en el entorno de desarrollo, que también es /var/www/html con el comando cd, y una vez allí ejecutamos el comando get index.html, siendo index.html el archivo a descargar, y tras ejecutarlo, nos descargará el archivo a la carpeta del entorno de explotación en la que habíamos ejecutado el comando SFTP, en este caso, /var/www/html. Por último usamos el comando exit para salir de la conexión con la máquina de desarrollo. También podemos subir archivos con el comando put.

```
sftp> cd /var/www/html
sftp> get index.html
Fetching /var/www/html/index.html to index.html
/var/www/html/index.html          100% 254    50.0KB/s   00:00
sftp> exit
```


También tenemos otra opción para transferir archivos entre los dos entornos, que sería conectarnos al de desarrollo con el FileZilla, descargar los archivos a la máquina anfitriona y posteriormente, conectarnos a la máquina de explotación y subir los archivos descargados anteriormente.