Material Web

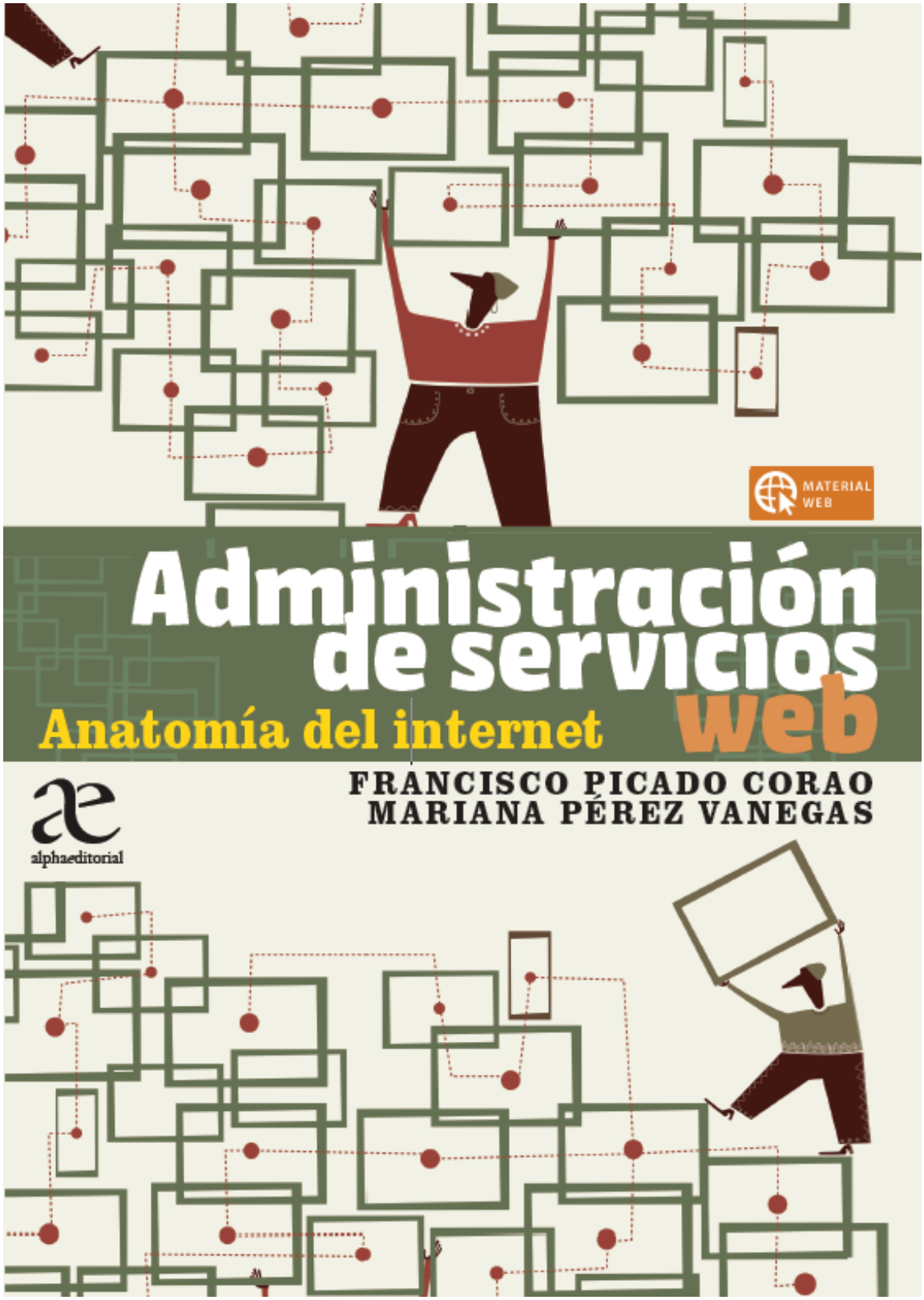


Tabla de contenido

Laboratorio 1. Configuración de un ambiente de red virtual en Oracle VM Virtual Box

Descripción. Este laboratorio permite enseñar al lector la manera de configurar máquinas virtuales en la herramienta *VirtualBox* y ponerlas en un entorno de red virtualizado, de manera tal que se puedan comunicar entre sí. Este laboratorio servirá como base a otros laboratorios que el lector encontrará a lo largo del libro. El software que se utiliza es gratuito.

Software necesario:

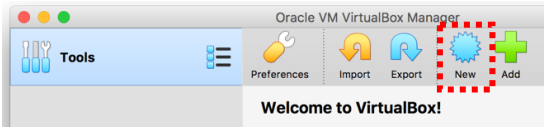
Links de descarga.

- Ubuntu 19.10 Desktop
  https://www.ubuntu.com/download/desktop
- Virtualbox 6.0.4
  https://www.virtualbox.org/wiki/Downloads
- Ubuntu Server 19.10
  https://www.ubuntu.com/download/server

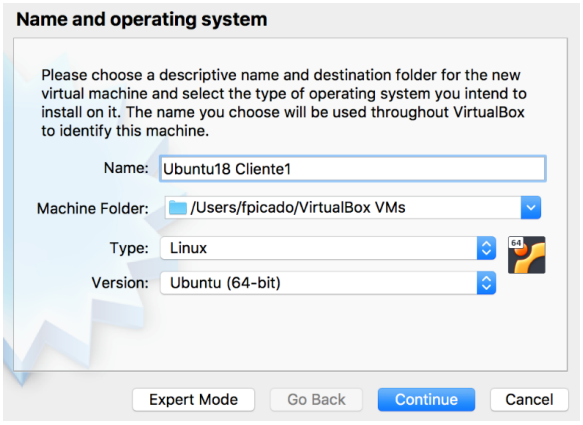
Creación de una maquina virtual nueva

Herramienta: Oracle VM VirtualBox

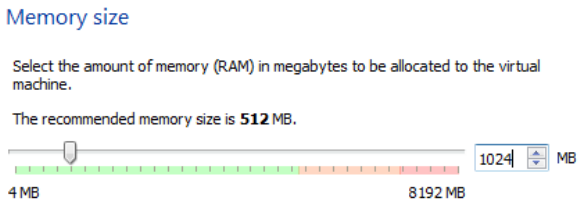
1. Abra VirtualBox.
2. En el panel de botones dar clic al botón **New**.



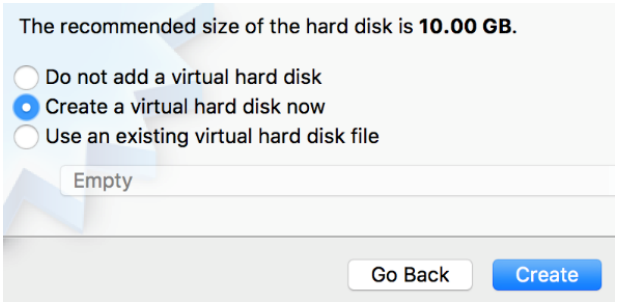
3. En la ventana que aparece, escribir el nombre para la máquina virtual: **Ubuntu19 Cliente1**, verifique el **tipo** y la **versión** y de clic en **Continue.**



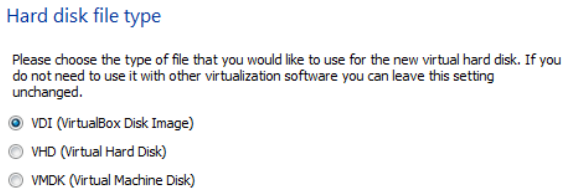
4. Dar a la VM un tamaño en memoria RAM de un Giga **(1024 MB)**, clic en **Continue**.



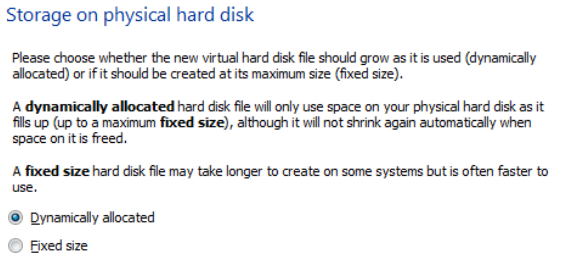
5. Crear ahora un nuevo **Virtual Disk** para la VM. Para eso elegir la opción **Create a Virtual Hard Disk Now**, luego clic en **Create.**



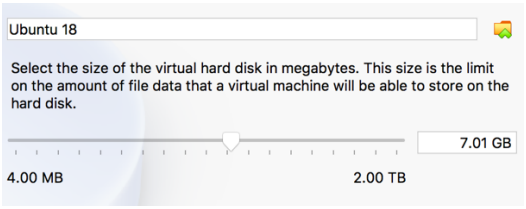
6. Elegir **VDI** ya que no se va a utilizar el disco virtual para ningún otro software de virtualización. Por ejemplo, **VMware Player,** clic en **Continue**.



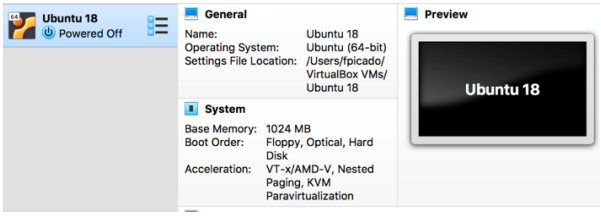
7. Seleccionar **Dynamically allocated** ya que esta opción no reserva el 100 % del espacio del disco virtual y por tanto, va consumiendo conforme se vaya necesitando. Clic en **Continue.**



8. Asignar los **10GBs** de espacio en disco que salen por defecto, clic en **Crear.**



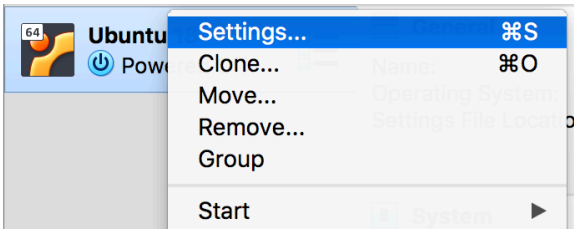
9. Ya con esto quedaría creada la VM. Sin embargo, el Sistema Operativo todavía no ha sido creado.



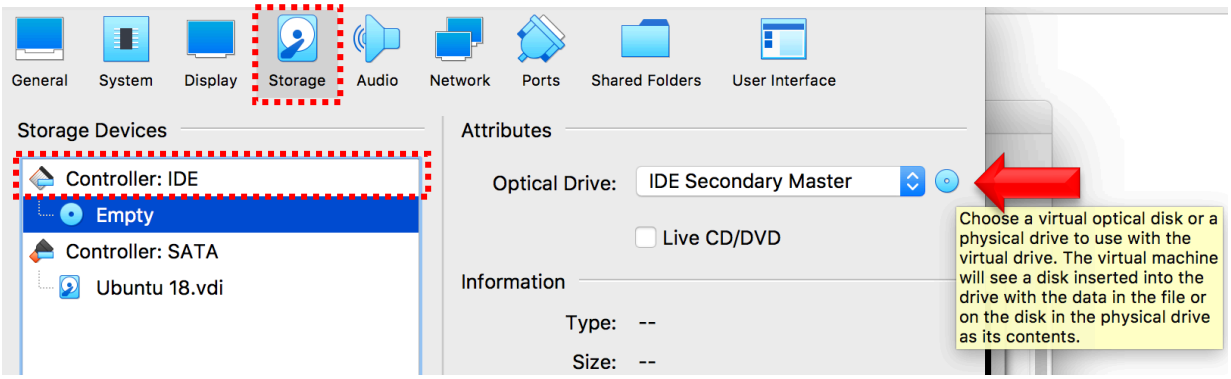
Instalación del Sistema Operativo

Una vez que la máquina virtual ha sido creada, el siguiente paso consiste en instalar el Sistema Operativo, en este caso será una versión de escritorio de Ubuntu 18.

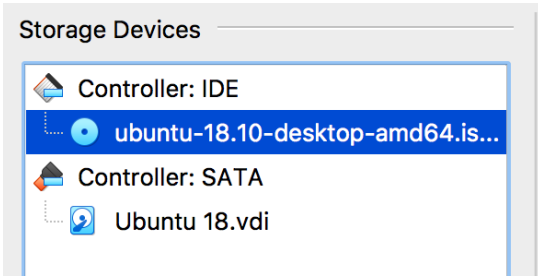
1. Dar clic derecho a la VM y elija la opción **Settings.**



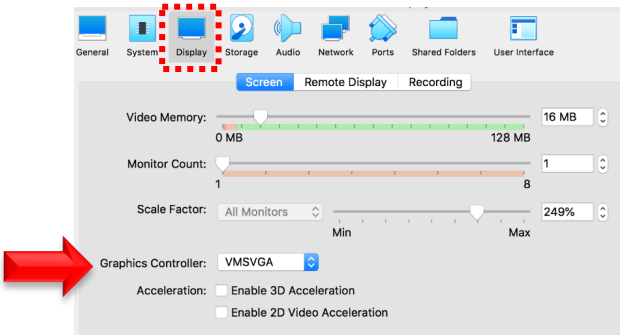
2. Elija la opción **Storage,** luego presione clic sobre el **CD** que sale **(Empty)** y luego de clic sobre el disco en **Optical Drive → Choose Virtual Optical Disk File.**



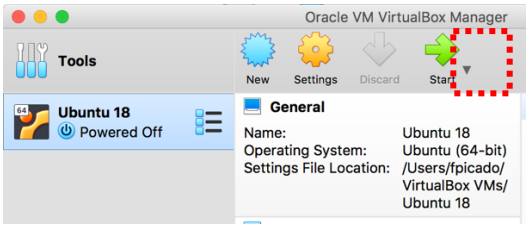
3. Elija la imagen .iso que se descargó al inicio y luego de clic sobre **OK,** una vez seleccionada la imagen, deberá verse de la siguiente manera**.**



4. Antes de instalar el Sistema Operativo, se aumentará el tamaño del monitor ya que, por defecto, el mismo sale muy pequeño y es difícil trabajar de esa manera. Para esto dará clic sobre **Settings → Display** y elegirá 200 % o más el **Scale Factor**. En la imagen, quedó de la siguiente manera.

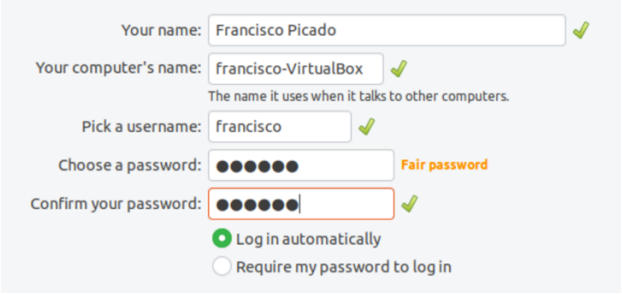


5. Ahora encenderás la VM y procederás a instalar el Sistema Operativo.

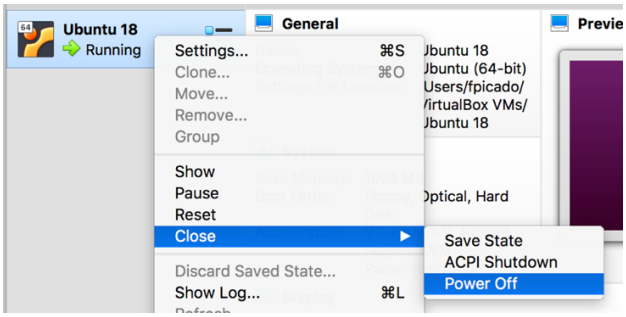


6. Para la instalación utilizar los siguientes parámetros. Las opciones podrían variar, pero esperamos que no mucho, igual lo importante es que Ubuntu quede instalado ☺.

- **Idioma de instalación:** English.
- En la parte derecha hay dos opciones, elegirán **Install Ubuntu.**
- **Keyboard layout:** El que salga por defecto y luego **Continue**.
- **Normal Installation** y quitar el check de **Download updates**, lo demás queda sin check y luego **Continue**.
- **Erase disk and install Ubuntu → Install Now**.
- **Write the changes to the disk?** Continue
- Ingresan sus datos y vayan hasta el final. En la sección de **Password**, elijan **Log in Automatically** para efectos de agilidad y una contraseña que no se le olvide (123456 funciona bastante bien).



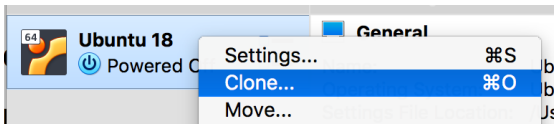
7. Una vez **finalizada la instalación**, apague la VM dando clic derecho sobre la misma y seleccionando las siguientes opciones.



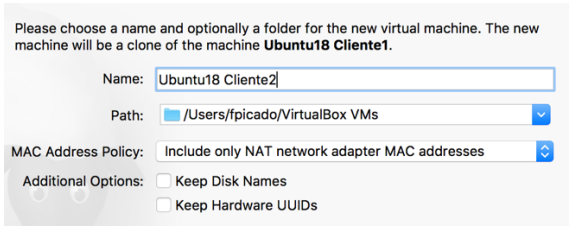
Clonar una Máquina Virtual

Para este laboratorio, se necesita tener dos clientes corriendo en el ambiente, para luego hacer las pruebas de la red virtual.

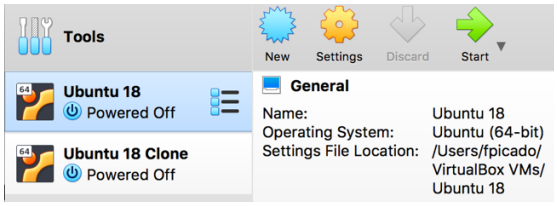
1. Clic derecho sobre la VM y elegir la opción **Clonar.**



2. Seleccionar los siguientes parámetros **(Ubuntu19 Cliente2)** y dar clic a **Siguiente**.



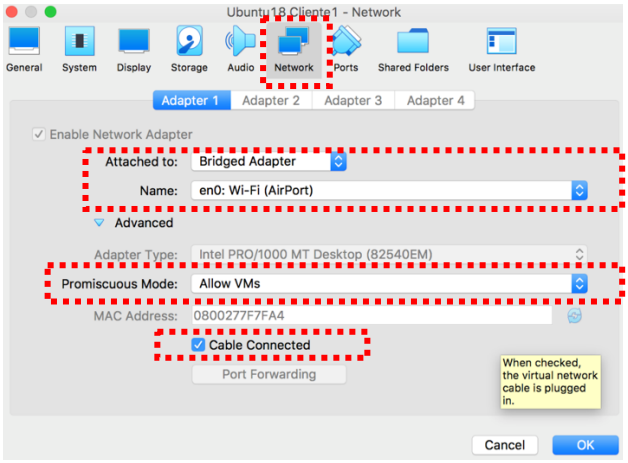
3. Seleccione la opción **Full Clone** y finalmente clic en **Clonar.**
4. Ahora deberás ver lo siguiente en el ambiente.



Configurar la red

Lo que sigue ahora es la configuración de la red para cada máquina virtual, para esto seguirás los siguientes pasos:

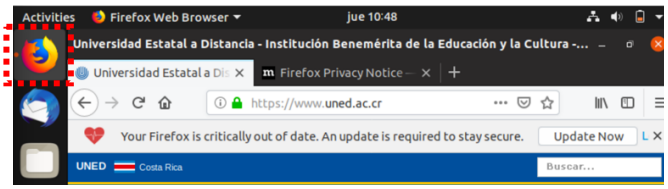
1. Apagar las máquinas virtuales.
2. Clic a la VM, luego clic en **Settings → Network.**
3. Una vez ahí, cambiar **Attached to** a **Bridge Adapter** y en **Name** deberás elegir la que diga Wi-Fi, en el ejemplo sale como **en0: Wi-Fi.**
4. La opción de **Promiscuous Mode** deberá decir **Allow VMs.**
5. **Cable Connect** con check.



Lo anterior deberá realizarse también para la segunda VM.

Verificar la conexión a internet

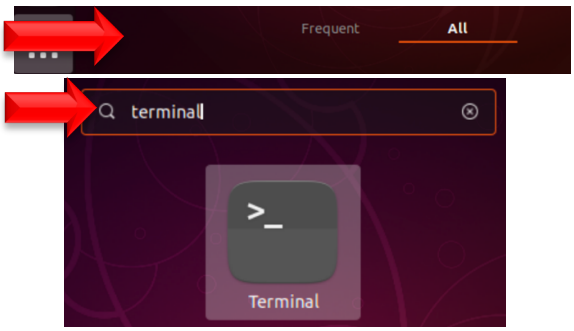
1. Encienda la máquina virtual **Ubuntu19 Cliente1** e ingresen a Mozilla y abra https://www.uned.ac.cr, si el sitio web abre, es porque hay conexión a Internet.



2. Haga lo mismo para la siguiente VM y verifique que tenga acceso a Internet.

Probar la interconectividad entre las VMs

1. Encienda cada máquina virtual y abra la terminal.



2. Luego digite el comando **ip addr show** y capture la dirección IP que sale como resultado de la consulta.

```
francisco@francisco-VirtualBox:~$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defau
lt qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP g
roup default qlen 1000
    link/ether 08:00:27:d4:9a:17 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.15/24 brd 192.168.0.255 scope global noprefixroute enp0s3
       valid_lft forever preferred_lft forever
    inet6 fe80::c265:7e88:8b15:3964/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Anote la IP Cliente 1 aquí _____**.**
Anote la IP Cliente 2 aquí _____**.**

3. Verifique que las dos IPs sean diferentes, pero en el mismo rango.
4. Desde la VM Cliente1 limpien pantalla con el comando **clear** y digite el comando **ping Cliente2_IP,** por ejemplo, en nuestro caso, la IP de Cliente2 es la **192.168.0.20**, entonces el comando que tenemos que digitar desde Cliente1 es **ping 192.168.0.20**, si la red se configuró correctamente, se verá algo así:

```
                                francisco@francisco-VirtualBox: ~
 File   Edit   View   Search   Terminal   Help
francisco@francisco-VirtualBox:~$ ping 192.168.0.20
PING 192.168.0.20 (192.168.0.20) 56(84) bytes of data.
64 bytes from 192.168.0.20: icmp_seq=1 ttl=64 time=0.518 ms
64 bytes from 192.168.0.20: icmp_seq=2 ttl=64 time=0.523 ms
64 bytes from 192.168.0.20: icmp_seq=3 ttl=64 time=0.482 ms
64 bytes from 192.168.0.20: icmp_seq=4 ttl=64 time=0.392 ms
64 bytes from 192.168.0.20: icmp_seq=5 ttl=64 time=0.468 ms
```

5. Realice lo mismo desde la VM Cliente2 pero haciendo ping a la IP del Cliente1.
6. Para detener el ping deberá dar **Ctrl+Z.**

Laboratorio 2. Inspección de un sitio web y transacciones en HTTP

Software necesario:

- Navegador de Internet Chrome, Safari, Mozilla.

Inspeccionar un sitio web

Cuando se solicita un sitio web, el protocolo HTTP es el que se encarga de ir al servidor web o servidor Origen y pedir el contenido que forma parte de este.

La primera vez, el servidor enviará el HTML de regreso al navegador y éste lo que hará será leer el código y comenzar a solicitar al Origen los objetos que se encuentre. Un sitio web puede componerse de los siguientes elementos.

- Texto
- Imágenes
- Archivos CSS
- Archivos JS
- Video o archivos multimedia
- Enlaces y demás

Entonces, cuando el navegador recibe el HTML, comienza a **parsear** las etiquetas y en el momento en que, por ejemplo, se encuentra con un CSS, irá al Origen, abrirá un nuevo canal de comunicación y pedirá el contenido, el servidor web lo enviará entonces y el navegador hará un **render** y lo presentará al usuario final.

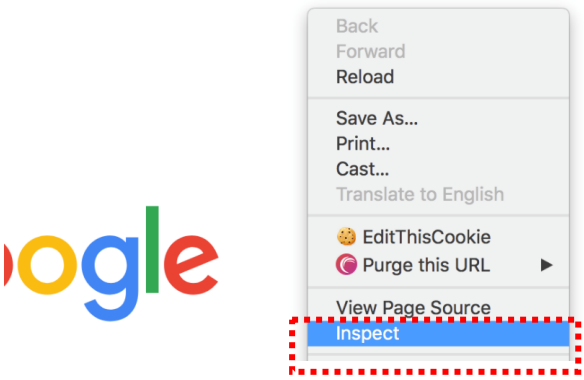
Todo este proceso en términos muy generales va a generar:

- **Métodos:** GET, POST, PUT, DELETE, HEAD, etc.
- **Códigos de respuestas.**
  100 Información, 200 Éxito (Success), 300 Redirección, 400 Errores de Usuario y 500 Errores de Servidor. Estos errores son muy útiles a la hora de diagnosticar un problema, ya que nos indican dónde se ubica el error o cual fue la secuencia hasta poder ver la página como tal.

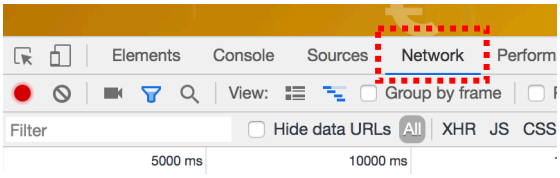
- **Encabezados de petición (request headers).**
  Es información que se envía con la petición del usuario al Origen.

- **Encabezados de respuesta (response headers).**
  Una vez que el servidor envía el contenido de vuelta al navegador, devolverá un código de respuesta y una serie de encabezados con diversidad de información. Para inspeccionar el sitio web, haremos lo siguiente.

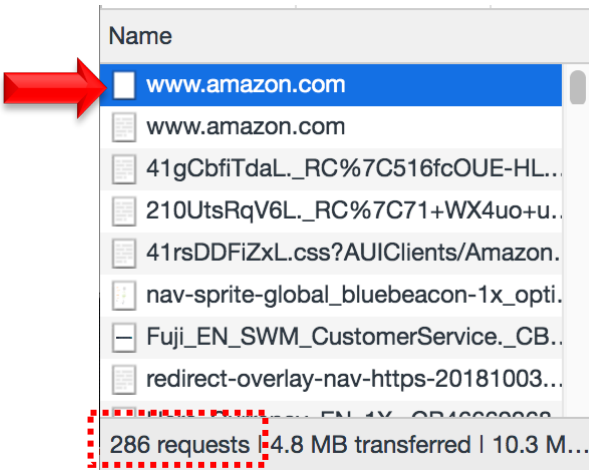
1. Abra Chrome, se recomienda hacer el laboratorio en **modo incógnito**.
2. En el área de contenido de clic derecho y seleccione **Inspeccionar.**



3. Digite en el espacio de la dirección http://www.amazon.com (verificar digitar http://), presione *enter* y espere que se abra la página, ahora de clic a **Network.**



4. Observe que se abre una lista de elementos. Estos elementos son los que el navegador tuvo que ir a solicitar al Origen. Seleccione el primero en la lista, el primero que dice www.amazon.com



Nota: si se observa la imagen anterior, HTTP tuvo que hacer **286 solicitudes** de **286 elementos** diferentes y todos forman parte de esa página que se desplegó en su navegador.

5. Una vez que dio clic sobre ese objeto, anote la siguiente información:

General Section.

- Request URL: _____
- Request Method: _____
- Status Code: _____

Response Headers.

- Location: _____

Request Headers.

- User-Agent: _____

6. Ahora, presione clic al segundo www.amazon.com y anote.

General Section.

- Request URL: _____
- Request Method: _____
- Status Code: _____

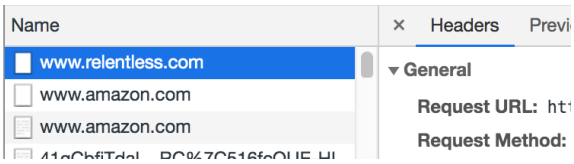
Response Headers:

- Cache-control: _____
- Content-encoding: _____
- Cache-language: _____

Request Headers:

- Accept-encoding: _____
- Accept-language: _____

7. Ahora digite en el navegador www.relentless.com (sin el http://) y de clic sobre el primer objeto de la lista.



Analice el objeto e indique que fue lo que sucedió, para esto base sus indicaciones en la sección General y el **Location** del Response Header.

_____

_____

_____

_____

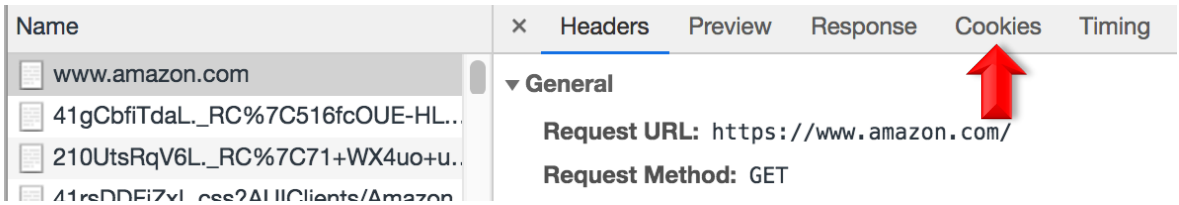
_____

_____

8. Ahora digite la siguiente dirección URL https://www.amazon.com/page/books.txt e indique qué código de respuesta obtuvo, busque en Google qué significa ese código de respuesta y anote la descripción al lado del código.

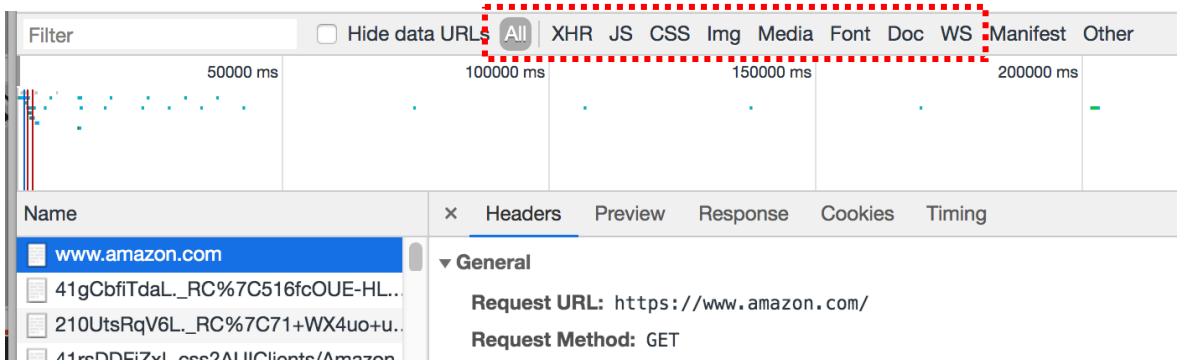
_____

9. Vuelva a digitar https://www.amazon.com, seleccione el www.amazon.com que sale en primer lugar de la lista, pero de clic sobre **Cookies** como se muestra a continuación.



¿Cuántos tipos de Cookies le salen? _____ No los nombres de las Cookies, despliegue toda la lista y analice qué tipos de Cookies le salen y anótelos acá: _____ y _____.

10. Para filtrar y ver qué tipos de objetos se solicitaron, se puede dar clic a las siguientes opciones:



11. Filtre por imágenes y seleccionen una de la lista, busquen en la sección de **Response Headers** y anoten:

- **Cache-control:** _____
- **Content-type:** _____

Códigos de respuesta

Busque en Internet y resuelva lo que se le solicita a continuación:
¿En qué consisten los siguientes códigos?

- **500:** _____
- **503:** _____
- **504:** _____
- **400:** _____
- **401:** _____
- **403:** _____

Laboratorio 3. Configuración de servidor web Apache

Requerimientos:

- Laboratorio 1 (Capítulo 1)

Software necesario:

- Ubuntu 18.10 Desktop
  https://www.ubuntu.com/download/desktop
- Virtualbox 6.0.4
  https://www.virtualbox.org/wiki/Downloads
- Ubuntu Server 18.04
  https://www.ubuntu.com/download/server

Apache es uno de los servidores web más utilizados en la actualidad junto con Ngix, siendo las dos, tecnologías importantes de conocer para cualquier desarrollador web.
El objetivo de este laboratorio es guiar al lector en el proceso de configuración de los servicios web de Apache en Ubuntu.

Antes de comenzar

1. Verificar que las máquinas virtuales que se crearon y configuraron en el laboratorio 1 estén apagadas.
2. Encender la máquina virtual Ubuntu19 Cliente1; este será el servidor web.

Instalar Apache

1. Digitar en la terminal el siguiente comando para actualizar el repositorio de versiones de software que se pueden actualizar o que pueden estar disponibles. Cada vez que se ejecuta alguna actualización en el sistema, Ubuntu o cualquier versión de Linux pedirá las credenciales de administrador de sistema:

sudo apt-get update

```
fpicado@fpicado:~$ sudo apt-get update
[sudo] password for fpicado:
Hit:1 http://archive.ubuntu.com/ubuntu cosmic InRelease
Get:2 http://archive.ubuntu.com/ubuntu cosmic-updates InRelease [88.7 kB]
Get:3 http://archive.ubuntu.com/ubuntu cosmic-backports InRelease [74.6 kB]
Get:4 http://archive.ubuntu.com/ubuntu cosmic-security InRelease [88.7 kB]
Get:5 http://archive.ubuntu.com/ubuntu cosmic/main Translation-en [513 kB]
Get:6 http://archive.ubuntu.com/ubuntu cosmic/restricted Translation-en [3,888 B]
Get:7 http://archive.ubuntu.com/ubuntu cosmic/universe Translation-en [5,063 kB]
Get:8 http://archive.ubuntu.com/ubuntu cosmic/multiverse Translation-en [113 kB]
Get:9 http://archive.ubuntu.com/ubuntu cosmic-updates/main Translation-en [102 kB]
Get:10 http://archive.ubuntu.com/ubuntu cosmic-updates/restricted Translation-en [2,192 B]
Get:11 http://archive.ubuntu.com/ubuntu cosmic-updates/universe Translation-en [84.4 kB]
Get:12 http://archive.ubuntu.com/ubuntu cosmic-updates/multiverse Translation-en [1,620 B]
Get:13 http://archive.ubuntu.com/ubuntu cosmic-backports/universe Translation-en [1,268 B]
Get:14 http://archive.ubuntu.com/ubuntu cosmic-security/main Translation-en [55.8 kB]
Get:15 http://archive.ubuntu.com/ubuntu cosmic-security/restricted Translation-en [2,192 B]
Get:16 http://archive.ubuntu.com/ubuntu cosmic-security/universe Translation-en [37.9 kB]
Get:17 http://archive.ubuntu.com/ubuntu cosmic-security/multiverse Translation-en [1,620 B]
Fetched 6,234 kB in 2s (3,950 kB/s)
Reading package lists... Done
```

A continuación, se instalará Apache mediante el siguiente comando, <digitar **Y** para confirmar>:
sudo apt-get install apache2

2. El servicio quedará instalado después de ejecutarse las instrucciones anteriores.

Probar la página por defecto de Apache

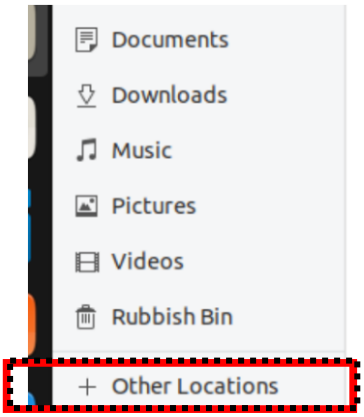
1. Encender la VM Cliente2.
2. Abrir Mozilla.

3. En la barra de navegación, digitar la IP de la VM Cliente1 <la IP dependerá de la red donde se encuentre el lector, en el laboratorio 1 se explica cómo encontrar las direcciones IP de las máquinas virtuales>, el resultado debería de ser el siguiente.

Así de sencillo se configura un servicio web en Apache.

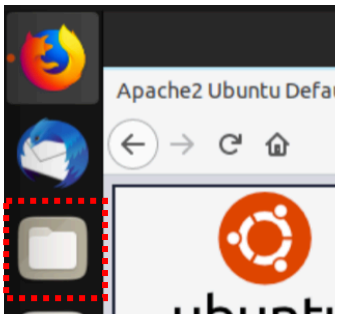
Navegar por la estructura de directorios

Para abrir el administrador de carpetas en Ubuntu, ir a la máquina virtual ***Cliente1*** y dar clic al siguiente ícono.

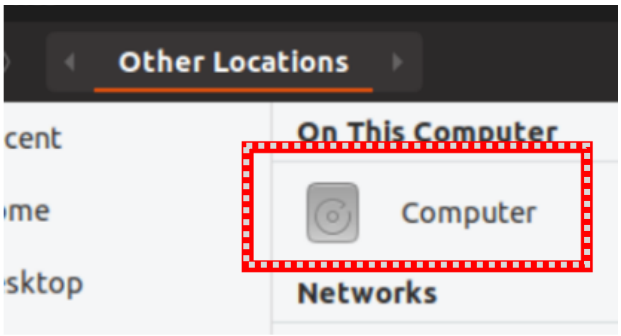


Nota: en el file manager de Ubuntu y algunas otras distribuciones de Linux no aparecen algunas carpetas que están ocultas. Para mostrar las carpetas ocultas y archivos ocultos presionar ctrl+h, observe la diferencia; para ocultar, hacer lo mismo.

Para ver la estructura de carpetas de Apache2 en Ubuntu, dar clic sobre la opción que dice + Other locations.

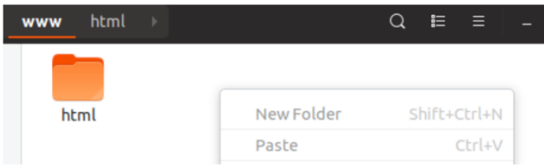


Luego doble clic sobre *Computer*.



Ahí se encontrará una carpeta que se llama var, el siguiente paso es ir a esta ubicación /var/www/html, donde se verá un archivo index.html, el cual es el html que se abre en el navegador de Cliente2 una vez instalado el Apache2.

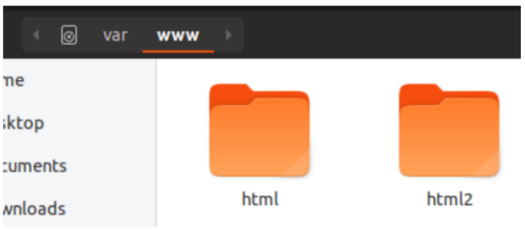
Dar doble clic sobre este archivo para confirmar que es el mismo archivo. El siguiente paso es posicionarse en la carpeta /var/www/, dar clic derecho y verificar si el sistema permite crear una nueva carpeta. La opción debería salir deshabilitada.



Crear un nuevo directorio con Nautilus

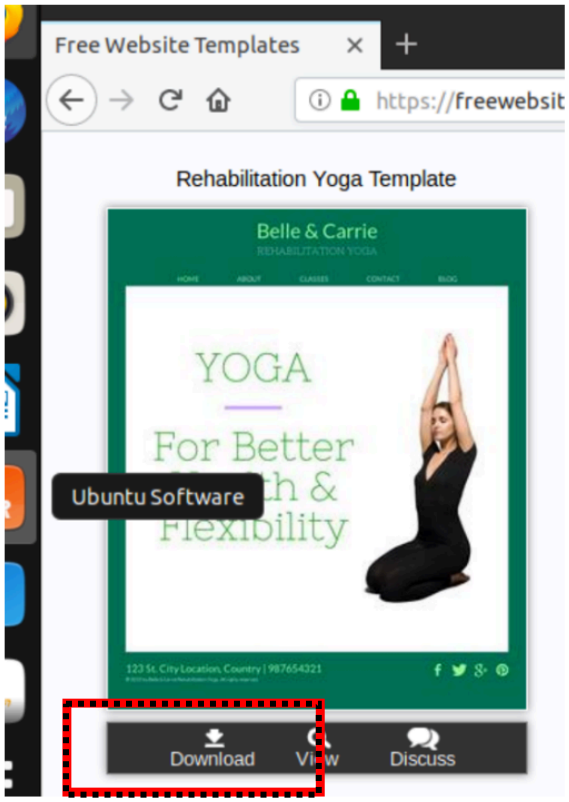
Nautilus es un administrador de archivos, el cual es el oficial para el proyecto GNOME y es el que permite crear carpetas cuando desde el administrador de archivos general no es posible.

Abrir la terminal y digitar el comando sudo nautilus. Se observa que se abre como otro administrador de archivos, dar clic sobre + Other locations y luego doble clic sobre Computer, ir a /var/www/ y seguidamente dar clic derecho, agregar entonces un nuevo directorio que se llame html2.

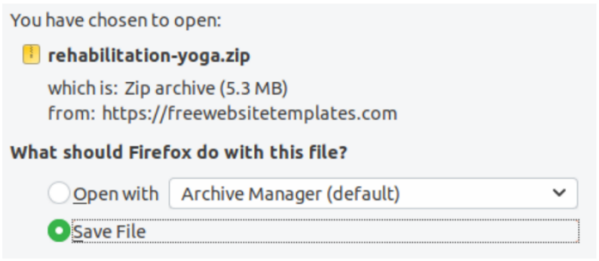


A este punto del laboratorio hay que verificar que las máquinas virtuales tengan acceso a Internet; si están bien configuradas como se explicó en el laboratorio 1, bastará con abrir el navegador y visitar cualquier sitio web, por ejemplo https://www.google.com. Si el sitio web abre, eso quiere decir que hay conexión.

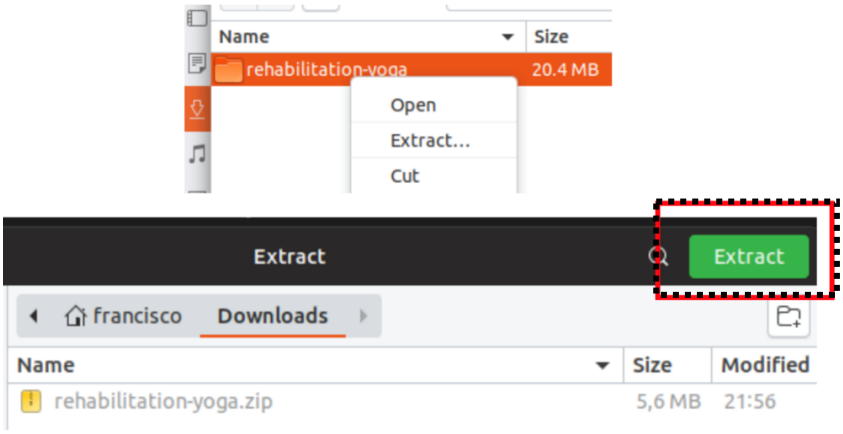
El siguiente paso es abrir la siguiente URL desde Mozilla. https://freewebsitetemplates.com/ y descargar el primer *template* que hay.



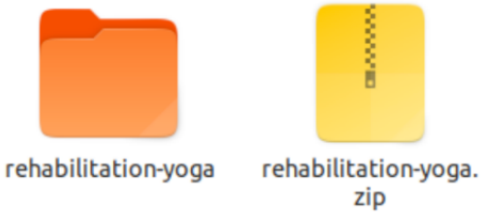
Seleccionar la siguiente opción.



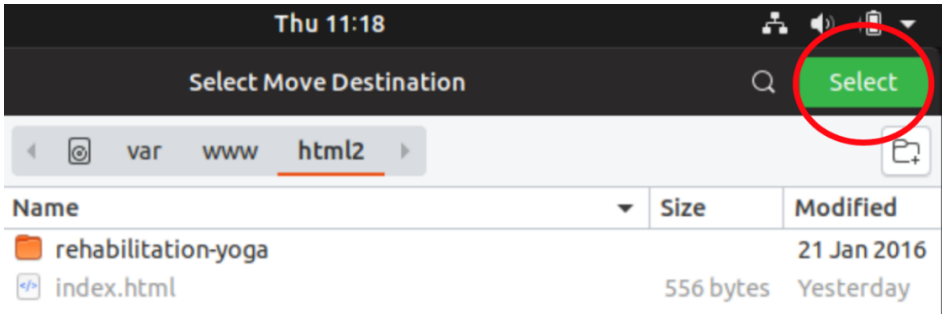
A continuación, ir a *Downloads*, dar doble clic al archivo .zip y luego clic derecho sobre la carpeta que aparece y elegir la opción *Extract*.



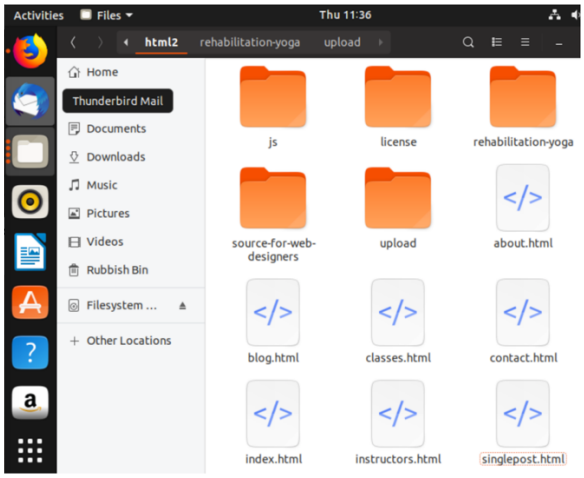
En *Downloads* ahora debería verse el archivo .zip y la carpeta descomprimida.



Dar clic derecho sobre la carpeta y elegir la opción Move to, seguidamente dar clic sobre + Other locations → var → www → html2; una vez ahí, dar clic sobre Select:



Nota: ignorar el index.html del *screenshot*.
Si ingresa a la carpeta *rehabilitation-yoga*, observará una carpeta *upload* y dentro de ella encontrará index.html. Mover el contenido de *upload* para la carpeta html2.



Editar el archivo de configuración

Para ir finalizando, ingrese nuevamente al Mozilla y digite la IP de su equipo para verificar que todavía esté cargando la página por defecto de Apache2.

¿Cómo es que esta IP está asociada a ese index en particular? Pues bien, desde la terminal de Cliente1 presione CTRL+Z y luego clear, y digite el siguiente comando:

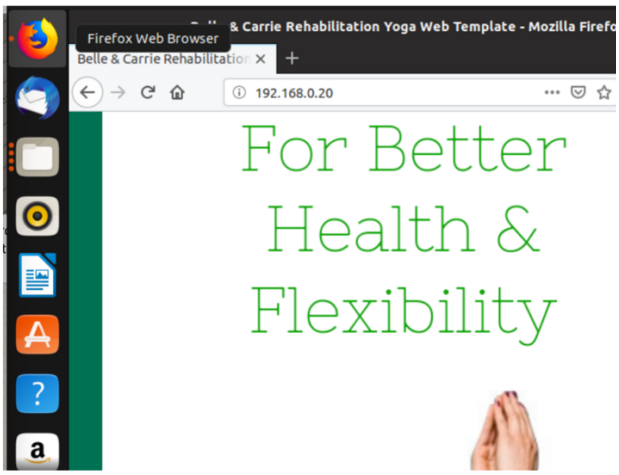
sudo nano /etc/apache2/sites-available/000-default.conf

En ese archivo que se abre es donde viene la configuración básica de Apache. Lo primero que se debe observar en la primera línea <VirtualHost *:80> es que, por ahora, esta página básica soporta solo HTTP y no HTTPS. Esto lo sabemos porque la versión segura de HTTP que es HTTPS utiliza puerto 443 y esto significa que por ahora no hay un certificado de seguridad.

Ir a la línea que dice DocumentRoot /var/www/html y modifíquelo de manera tal que se lea DocumentRoot /var/www/html2 para luego presionar Ctrl+X, digitar Y y presionar Enter. Finalmente, reiniciar el servicio de Apache corriendo el comando sudo systemctl restart apache2.

Ahora ir a Cliente2, el navegador debería estar todavía abierto, entonces dar clic en refrescar a la página o digite la IP de Cliente1 nuevamente, ya la página por defecto no debería salir, en su lugar debería de salir el nuevo sitio web.

Este archivo de configuración es el que maneja la ubicación del index, archivos .conf puede haber tantos como sitios web se tengan hospedados.



Ahora navegue por el sitio para comprobar su funcionalidad.

Laboratorio 4. Instalar un certificado *Single-Sign* para tráfico seguro

Antes de comenzar

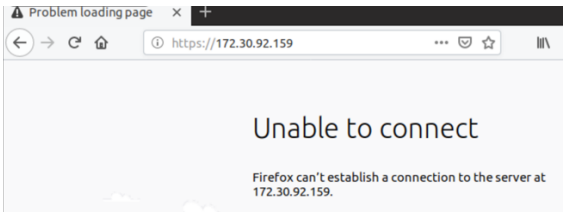
1. Verificar que el laboratorio 3 ha sido completado.
2. Verificar que las dos máquinas virtuales estén apagadas.
3. Clonar Cliente1 y asignarle el nombre **Ubuntu19 Cliente1 Single-Sign Cert**. Esto a manera de backup para no perder lo hecho hasta ahora.

III. Creando el certificado de Seguridad

La configuración que se ha hecho hasta ahora corre sobre tráfico no seguro (HTTP) ya que Apache por defecto envía tráfico sobre el puerto 80.

La idea de este taller es instalar un certificado sencillo **(Single-Sign Certificate)** y realizar las pruebas para verificar que el mismo funciona y que el tráfico pasará entonces por HTTPS.

1. Encender la nueva VM.
2. Entrar al navegador Mozilla y digitar la siguiente dirección URL https://ip_desuVM.
3. A este punto debería de aparecer un error como el siguiente.



4. Lo anterior sucede a lo dicho anteriormente, el servicio de Apache no permite tráfico seguro porque para esto se requiere de un certificado.
5. Abrir la terminal y correr el siguiente comando que se emplea para crear un certificado nuevo**: sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt**
6. Una vez dentro, ingrese la siguiente información:
   Output
   Country Name (2 letter code) [AU]:**CR**
   State or Province Name (full name) [Some-State]:**San Jose**
   Locality Name (eg, city) []:**Sabanilla**
   Organization Name (eg, company) [Internet Widgits Pty Ltd]:**UNED**
   Organizational Unit Name (eg, section) []:**Talleres**
   Common Name (e.g. server FQDN or YOUR name) []:**talleresuned.uned.com**
   Email Address []:**su_correo_electronico**
7. Una vez hecho lo anterior, el certificado estará listo para ser utilizado.

II. Configurar Apache para utilizar SSL

En la primera sección de este laboratorio, se generó la llave y los archivos del certificado en el directorio **/etc/ssl**, lo que sigue es modificar la configuración del Apache para poder hacer uso de HTTPS.

1. Limpie pantalla: **clear**.
2. Ejecute el comando **sudo nano /etc/apache2/conf-available/ssl-params.conf** y en el archivo en blanco digite o pegue el siguiente código:

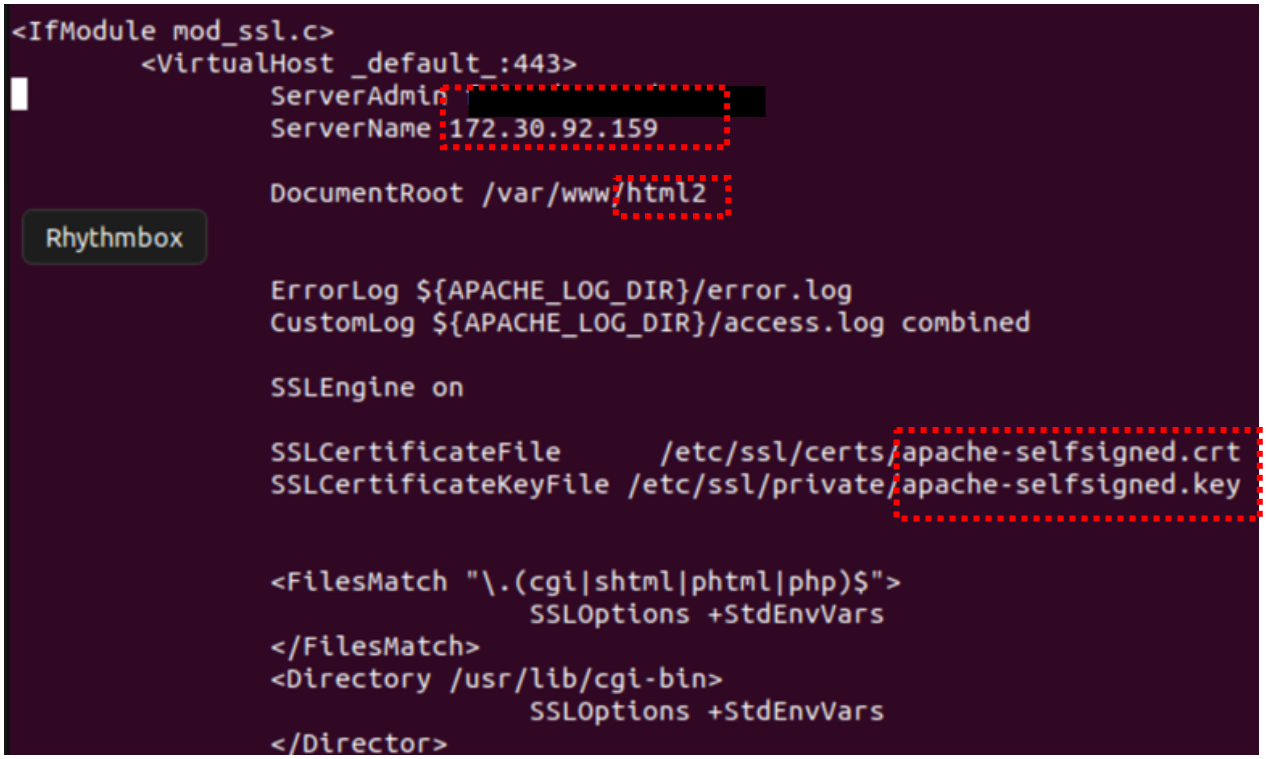
   SSLCipherSuite EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH
   SSLProtocol All -SSLv2 -SSLv3 -TLSv1 -TLSv1.1

SSLHonorCipherOrder On
Disable preloading HSTS for now. You can use the commented out header line that includes
the "preload" directive if you understand the implications.
Header always set Strict-Transport-Security "max-age=63072000; includeSubDomains; preload"
Header always set X-Frame-Options DENY
Header always set X-Content-Type-Options nosniff
Requires Apache >= 2.4
SSLCompression off
SSLUseStapling on
SSLStaplingCache "shmcb:logs/stapling-cache(150000)"
Requires Apache >= 2.4.11
SSLSessionTickets Off

3. Guarde los cambios precionando **Ctrl+X, Y** y finalmente presione **Enter.**

III. Modificar el SSL virtual host file de Apache

1. El siguiente paso es modificar **default-ssl.conf** pero antes es importante hacer una copia de seguridad ejecutando el comando: **sudo cp /etc/apache2/sites-available/default-ssl.conf /etc/apache2/sites-available/default-ssl.conf.bak**
2. Ir al archivo y hacer los ajustes necesarios: **sudo nano /etc/apache2/sites-available/default-ssl.conf**
3. El archivo que se abre es algo extenso porque viene con muchos comentarios, pero revisando cuidadosamente incluir la siguiente información (utilizando la información personal del lector).

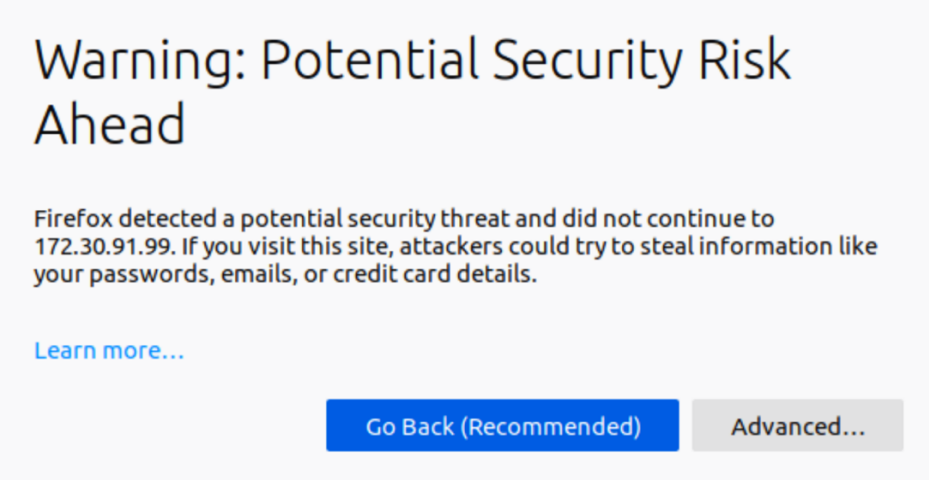


IV. Habilitar los cambios en Apache

1. Verificar que el firewall esté deshabilitado: **sudo ufw status**, si el resultado dice **Status: inactive** entonces ir al paso #2.
2. El siguiente paso es habilitar los cambios efectuados hasta ahora y para eso habrá que habilitar los módulos de SSL y los encabezados respectivos corriendo los comandos:
   sudo a2enmod ssl
   sudo a2enmod headers
3. El siguiente comando lo que permite es habilitar el SSL virtual host: **sudo a2ensite default-ssl**
4. El siguiente paso consiste en habilitar los cambios que se hicieron en ssl-params.conf: **sudo a2enconf ssl-params**
5. Hasta este momento ya se tienen los módulos necesarios habilitados, entonces lo que sigue es revisar que no existan errores: **sudo apache2ctl configtest**
   Si se despliega el siguiente resultado, es porque todo se ha configurado correctamente hasta este momento:
   AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this message
   Syntax OK
6. El último paso es reiniciar el servicio de apache2: **sudo systemctl restart apache2**

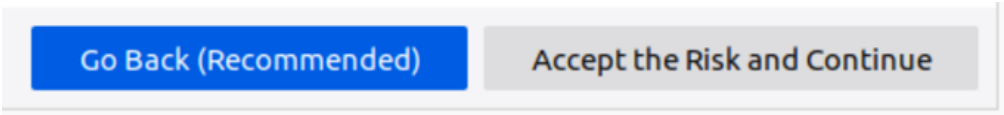
V. Probar la configuración

1. Abrir el navegador y digitar la ip usando http, esto para verificar que la página esté abriendo.
2. Ahora, agregar https:// al inicio y verificar si se despliega la siguiente alerta:



Lo anterior es totalmente esperado ya que este certificado que ha sido creado todavía no ha sido aprobado o firmado por ningún CA o Certificate Authority que es otra parte del procedimiento, pero en este caso, de igual manera la información viajará de forma encriptada.

3. Dar clic **Advanced** y luego **Accept the risk and continue:**



4. Una prueba adicional que se puede realizar es ir a la máquina virtual del cliente 2 y acceder desde ahí la página haciendo uso de https.

VI. Redirección

Se sabe que una solicitud de HTTP envía información importante al servidor por medio de los headers o encabezados y que el servidor sirve los objetos también con información de respuesta y además con un código de estado.
Los 300s son códigos de respuesta para redirecciones, el más común es el 301 que significa que el objeto ha sido movido permanentemente.

Lo siguiente en este ejercicio es forzar el protocolo para que la solicitud se haga por medio de https, eso quiere decir que si el usuario digita http://www.example.com, entonces que se haga una redirección a https://www.example.com, algo que se practicó en el laboratorio 3.

1. Ingresar a la edición del archivo **000-default.conf: sudo nano /etc/apache2/sites-available/000-default.conf**
2. Una vez ahí, insertar las siguientes entradas, utilizando la IP de la máquina virtual:

```
ServerAdmin webmaster@localhost
ServerName 172.30.91.99
DocumentRoot /var/www/html2


Redirect / https://172.30.91.99/
```

3. Salve y cierre la edición **(CTRL+X,** luego **Y** y **Enter)** y finalmente reinicie apache **sudo systemctl restart apache2.**
Para evitar problemas de caching, se recomienda reiniciar la máquina virtual y ahora sí, ir al navegador y digitar únicamente la IP, como resultado se debería poder observar que la dirección cambia a https:// e inclusive si se hace la inspección se podrá observar que esta acción genera un 302.

Solucionario de ejercicios

Capítulo 1

1. C
2. A
3. C
4. B
5. A
6. A
7. B
8. B
9. C
10. D
11. C
12. D
13. C
14. B
15. D
16. C
17. A
18. C
19. D
20. A
21. D
22. B

Capítulo 2

1. B
2. A y B
3. A, B y D
4. A
5. C
6. D
7. B, C, D
8. B
9. C
10. A, B Y D

Capítulo 3

1. A, C
2. A, B, C, D
3. B, C, D
4. B, C, D
5. A, B, C
6. A, C, D
7. C
8. A
9. B
10. A, B, D

Capítulo 4

1. Plataforma
2. Software
3. Balanceador
4. Infraestructura
5. Microsoft
6. Pública
7. Kerberos
8. Origen
9. Escalabilidad
10. Radius

Capítulo 5

DNS y HTTP

(3)	PTR
(6)	Incremental
(1)	Puerto 53
(7)	Puerto 443
(2)	Root
(9)	Recursiva
( )	Iterativa
(11)	CNAME
( )	UDP
(5)	TCP
( )	TLD
(13)	Autoritario
(10)	SOA
(4)	Puerto 80
(14)	TTL
( )	AAAA
(12)	Full
(8)	Server push

Capítulo 6

Horizontales

1. Etag
2. Cache-control
3. Privado
4. No-cache
5. Deflate
6. Minificar
7. Max age

Verticales

1. Public
2. Caching
3. No-Store
4. CDN
5. Expires

Capítulo 7

1. A
2. A
3. A, B, C, D
4. A, B, D
5. A, B, C
6. D
7. A
8. A, B, C, D
9. B
10. D