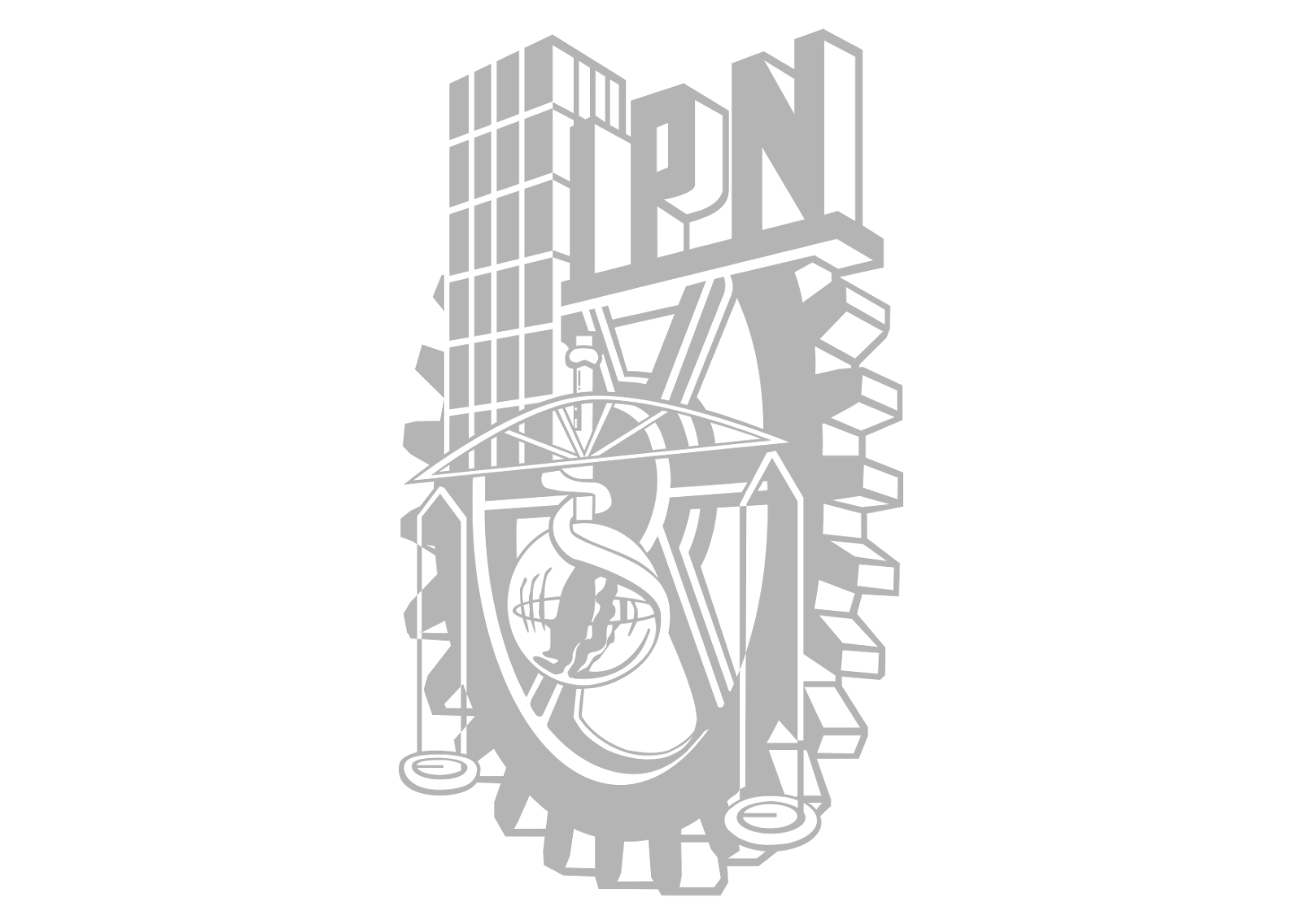
**Instituto Politécnico Nacional**

**Escuela Superior de Cómputo**

Administración de Servicios en Red

Equipo 9

**Servidor de aplicación utilizando el protocolo HTTP: manual de instalación/configuración/operación de servidor.**

Rocío Vivia Delgadillo

Bryan Domínguez de la Rosa

Fernando Jair Pacheco Díaz

Grupo 4CV3

Profesor: Manuel Alejandro Soto Ramos

**Índice**

Protocolo HTTP -------------------------------------------------------------------- 3

Servidor Apache ------------------------------------------------------------------- 3

Instalación -------------------------------------------------------------------------- 3

Configuración ----------------------------------------------------------------------- 3

Operación --------------------------------------------------------------------------- 5

Contenedor virtual por IP y por dominio ----------------------------------- 5

Configuración de puerto de operación-------------------------------------- 6

Servidor de aplicación utilizando el protocolo HTTPS---------------------- 8

Personalización de páginas de error para todos los sitios------------------ 13

Configuración de archivos de bitácoras y mensajes de error--------------- 14

Resumen de operación de forma dinámica --------------------------------- 15

Referencias ------------------------------------------------------------------- 18

**Protocolo HTTP**

El Protocolo de Transferencia de HiperTexto (Hypertext Transfer Protocol) es un sencillo protocolo cliente-servidor que articula los intercambios de información entre los clientes Web y los servidores HTTP. La especificación completa del protocolo HTTP 1/0 está recogida en el RFC 1945.

El desarrollo de HTTP fue iniciado por Tim Berners-Lee en el CERN en 1989. El desarrollo de estándares de HTTP fue coordinado por Internet Engineering Task Force (IETF) y World Wide Web Consortium (W3C), culminando en la publicación de una serie de Requests for Comments (RFCs).

**Servidor Apache HTTP versión 2.4**

El Proyecto Apache HTTP Server es un esfuerzo para desarrollar y mantener un servidor HTTP de código abierto para los sistemas operativos modernos, incluidos UNIX y Windows. El objetivo de este proyecto es proporcionar un servidor seguro, eficiente y extensible que proporcione servicios HTTP en sincronización con los estándares HTTP actuales.

El Servidor Apache HTTP ("httpd") se lanzó en 1995 y ha sido el servidor web más popular en Internet desde abril de 1996. En febrero de 2015, celebró su vigésimo aniversario como proyecto.

**Instalación en Ubuntu 16.04 LTS**

Actualizamos e instalamos el servidor escribiendo las siguientes líneas en consola:

$sudo apt-get update

$sudo apt-get upgrade

$sudo apt install apache2

Una vez instalado el servidor, iniciamos el servicio instroduciendo la siguiente línea en consola:

$sudo service apache2 start

Verificamos que el servicio haya iniciado realmente con la siguiente línea:

$sudo systemctl status apache2

Nota: Para detener el servicio basta con cambiar “start” por “stop”. Existen otras opciones como: reload, restart, status.

**Configuración**

Con permisos de superusuario, creamos las carpetas contenedoras de la información que los sitios web requieren:

$sudo su

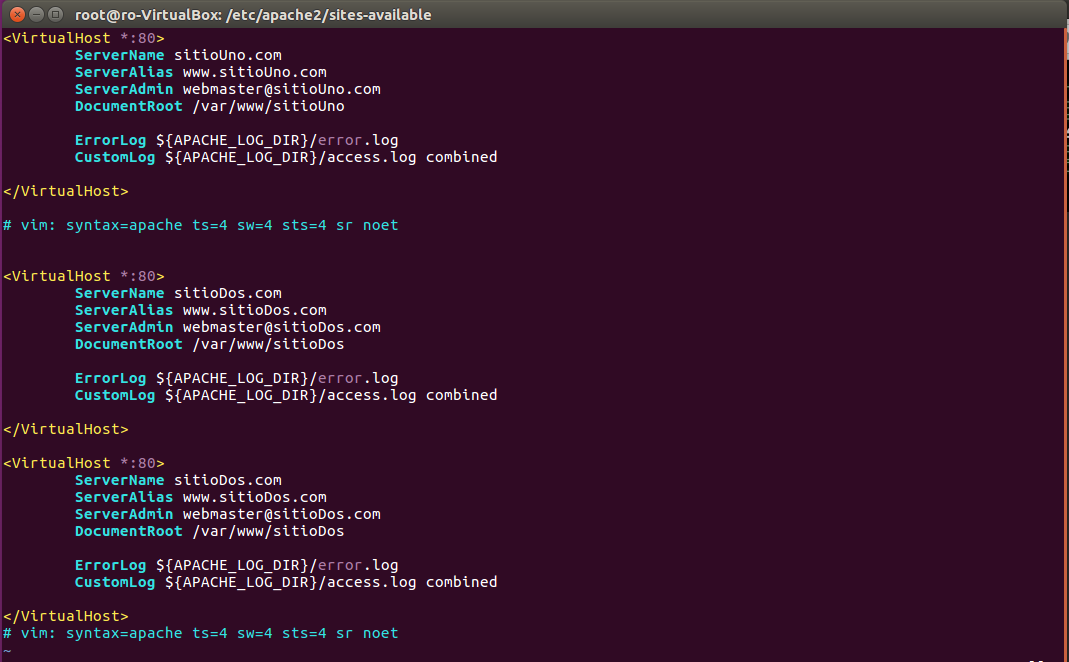
#mkdir /var/www/sitioUno

Lo siguiente será configurar los VirtualHost, para ello, nos movemos a la carpeta /etc/apache2/sites-avaiable/ y crearemos el archivo de configuración para los hosts, copiaremos el que tenemos por default:

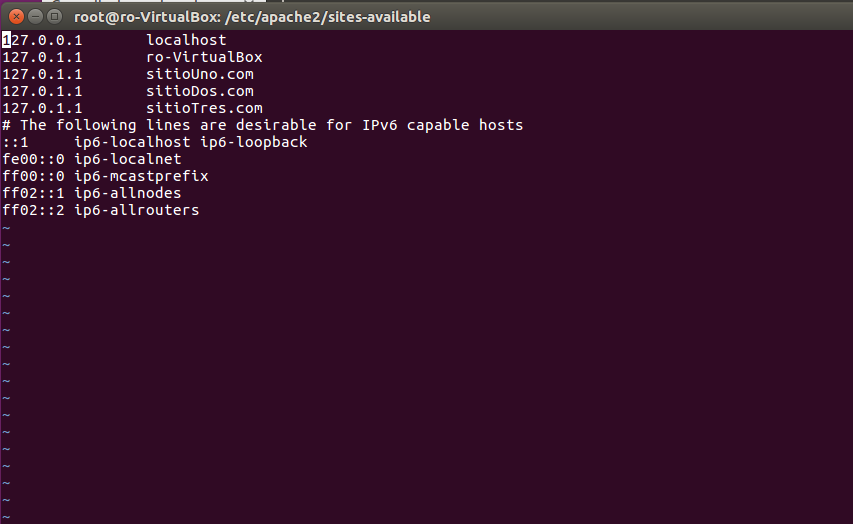
#cd /etc/apache2/sites-available/

#cp 000-default.conf sitioUno.conf

Una vez copiado el archivo .conf, podemos modificar el nombre, alias, tenemos que proporcionar la ruta de los sitios y podemos también cambiar el correo electrónico:



Agregamos las direcciones de los hosts con sus respectivos nombres de dominio en el archivo /etc/hosts de la siguiente manera:



Por último, deshabilitamos el archivo host por default y habilitamos el nuevo:

#a2ensite sitioUno.conf

#a2dissite 000-default.conf

#service apache2 reload

**Operación**

1. **Contenedor virtual por IP y por dominio**
   1. Restringir acceso al recurso por dirección IP del cliente: entramos al archivo /etc/apche2/apache2.conf y agregamos las directivas.

<Directory (ruta del directorio)>

Options All AllowOverride All <RequireAll> Require all granted Require ip x.x.x.x #IP a la que se le niega acceso. </RequireAll> </Directory>



* 1. Restringir acceso al recurso por segmento de red: entramos al archivo /etc/apache2/apache2.conf y agregamos las directivas:

<Directory (ruta del directorio)> Options All AllowOverride All <RequireAll> Require all granted Require ip x.x.x.x x.x.x.x #Segmento de red. </RequireAll> </Directory>

* 1. Restringir acceso al recurso por nombre de usuario (grupo de usuarios) /clave de acceso: aquí definiremos un grupo de usuarios que tienen los permisos para acceder al sitio web, para ello primero necesitaremos crear el archivo /etc/apache2/htpasswd:

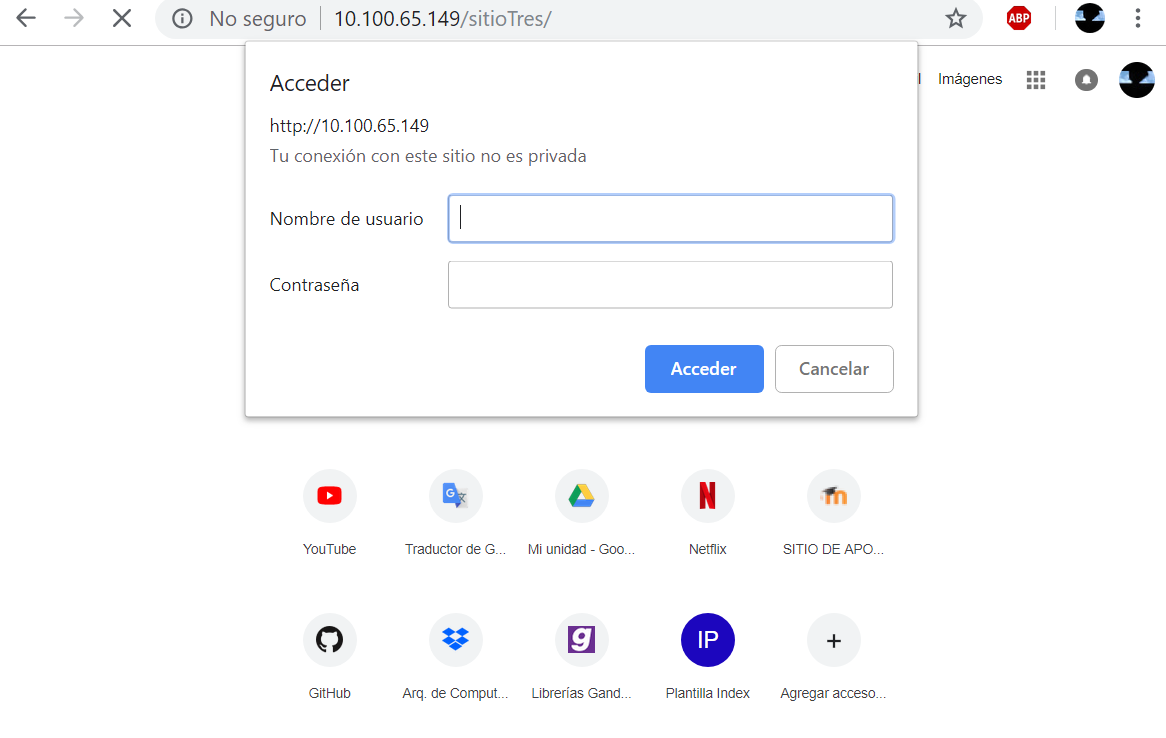
# htpasswd -c /etc/apache2/.htpasswd “usuario”

Y a continuación nos pedirá ingresar la contraseña. Para consultar la lista de usuarios basta con acceder al archivo .htpasswd y también podemos observar que las respectivas contraseñas aparecen de forma cifrada.

Una vez creado el archivo de usuarios, procedemos a restringir el recurso por nombre de usuario:

<Directory (ruta del directorio)> Options All AllowOverride All <RequireAll> AuthName ”Private” AuthType Basic AuthUserFile /etc/apache2/.htpasswd Require valid user </RequireAll> </Directory>

Utilizamos **AuthType** Basic indicando que el método de autenticación es básico, **AuthFile** para indicar la ruta del archivo que define los usuarios válidos.

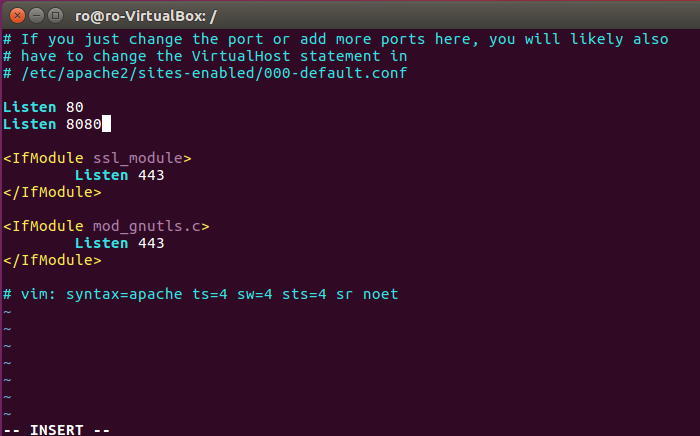


1. **Configuración de puerto de operación**

La configuración de los puertos se realiza en el archivo /etc/apache2/ports.conf, el puerto que se usa por defecto es el 80, pero para agregar un basta con entrar al archivo y agregar la siguiente línea:

# vim /etc/apache2/ports.conf

Listen 80 Listen puertoNuevo

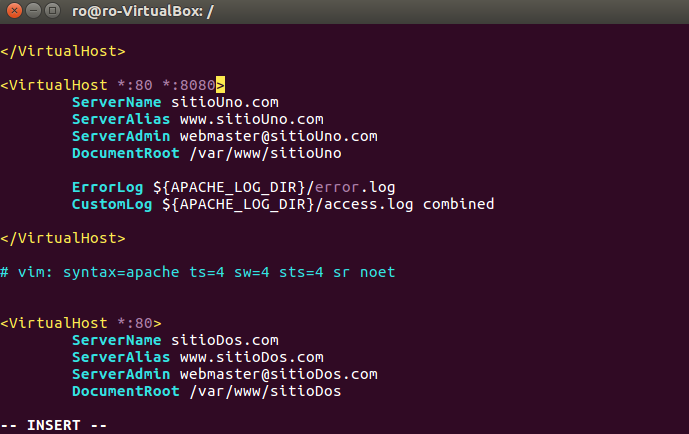


Lo siguiente sería agregar en los VirtualHost el puerto nuevo:

<VirtualHost \*:80 \*:puertoNuevo>

...

</VirtualHost



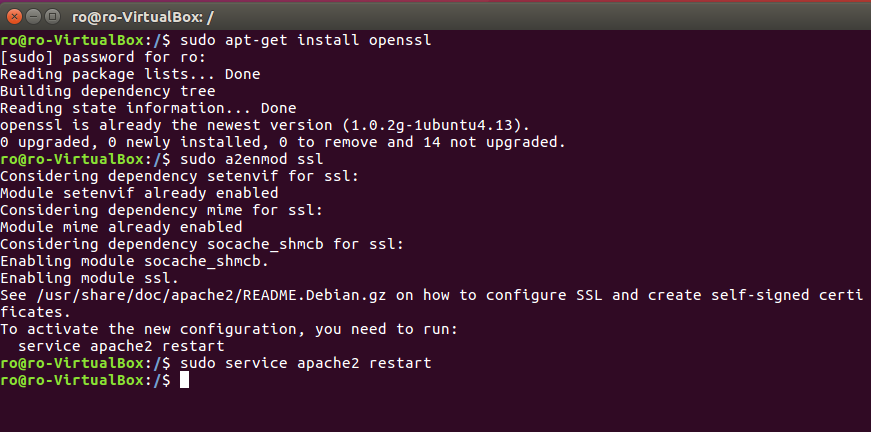
Por último, reiniciamos el servidor:

#service apache2 restart

1. **Servidor de aplicación utilizando el protocolo HTTPS**
   1. Definición de certificados /llaves de operación

Necesitamos tener instalado openssl, un Ubuntu 16.04 LTS lo está, para confirmar podemos intentar instalarlo con el siguiente comando:

$sudo apt-get install openssl



Habilitamos el modulo introduciendo la siguiente línea en la terminal y reiniciamos el servidor:

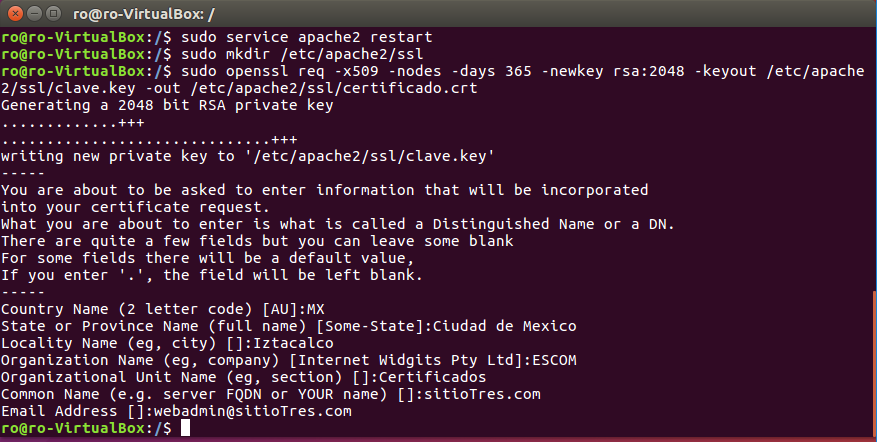
$sudo a2enmod ssl $sudo service apache2 restart

Lo siguiente es crear una carpeta destino para los certificados que crearemos en un momento

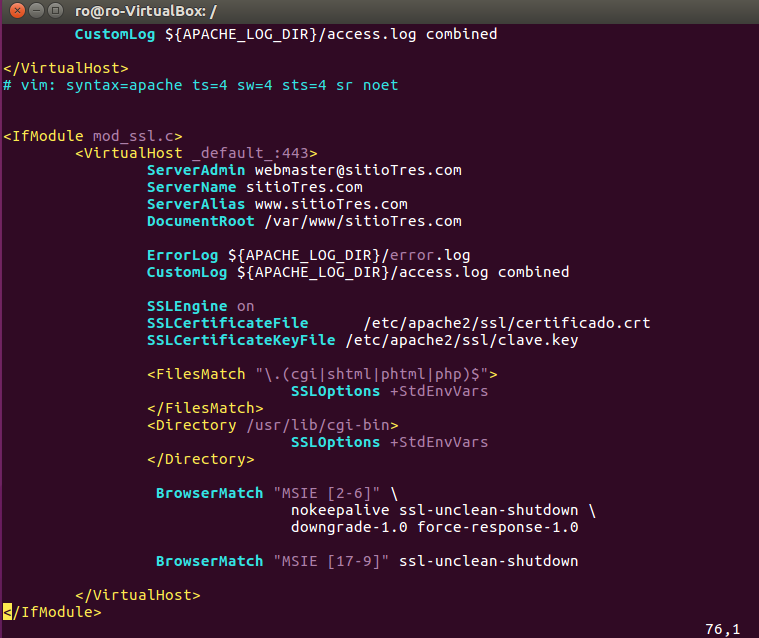
$sudo mkdir /etc/apache2/ssl

Creamos el archivo de certificado y la clave, para ello introducimos la siguiente línea de comando:

$sudo openssl req -x509 -nodes -days 365 -netke rsa:2048 -keyout /etc/apache2/ssl/clave.key -out /etc/apache2/ssl/certificado.crt



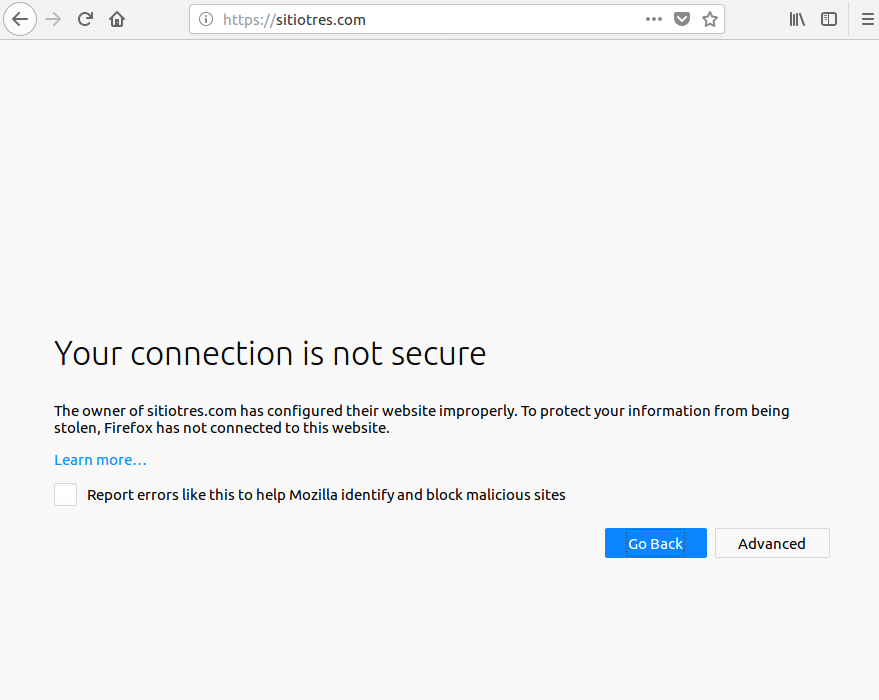
Una vez introduciendo la información requerida, ahora configuramos apache para hacer uso de los certificados ya creados, para ello nos ubicamos en el archivo /etc/apache2/sites-available/sitioUno.conf y agregamos la siguiente información:



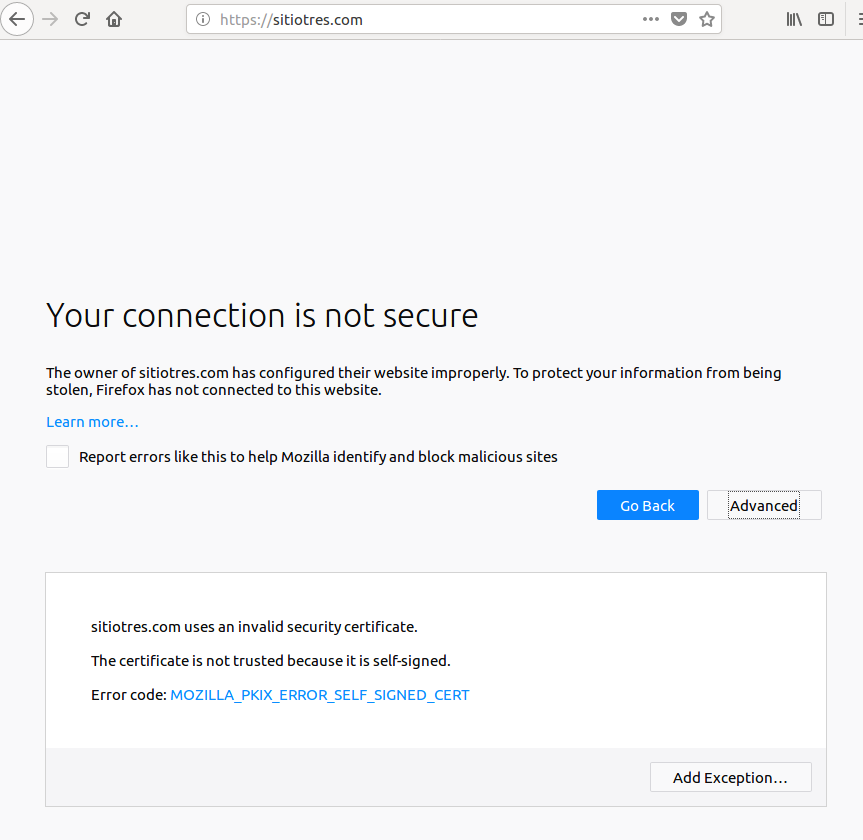
Y recargamos nuevamente el servidor

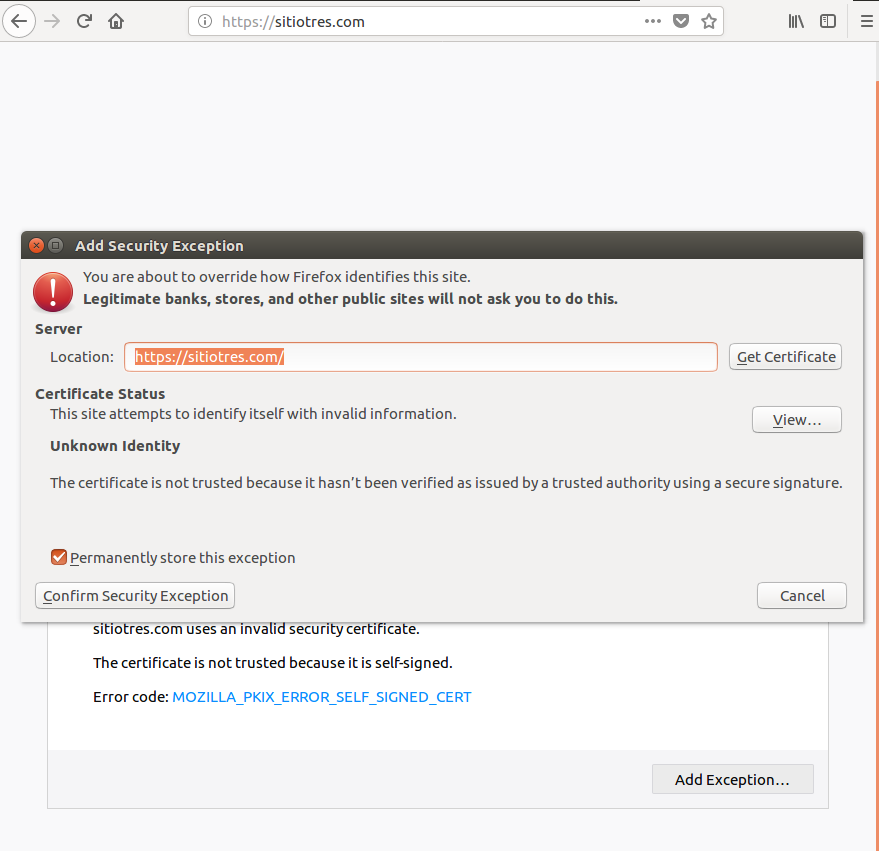
$sudo service apache2 reload

Para finalizar, abrimos una ventana de navegación y escribimos en la barra de direcciones https://sitioCertificado.com

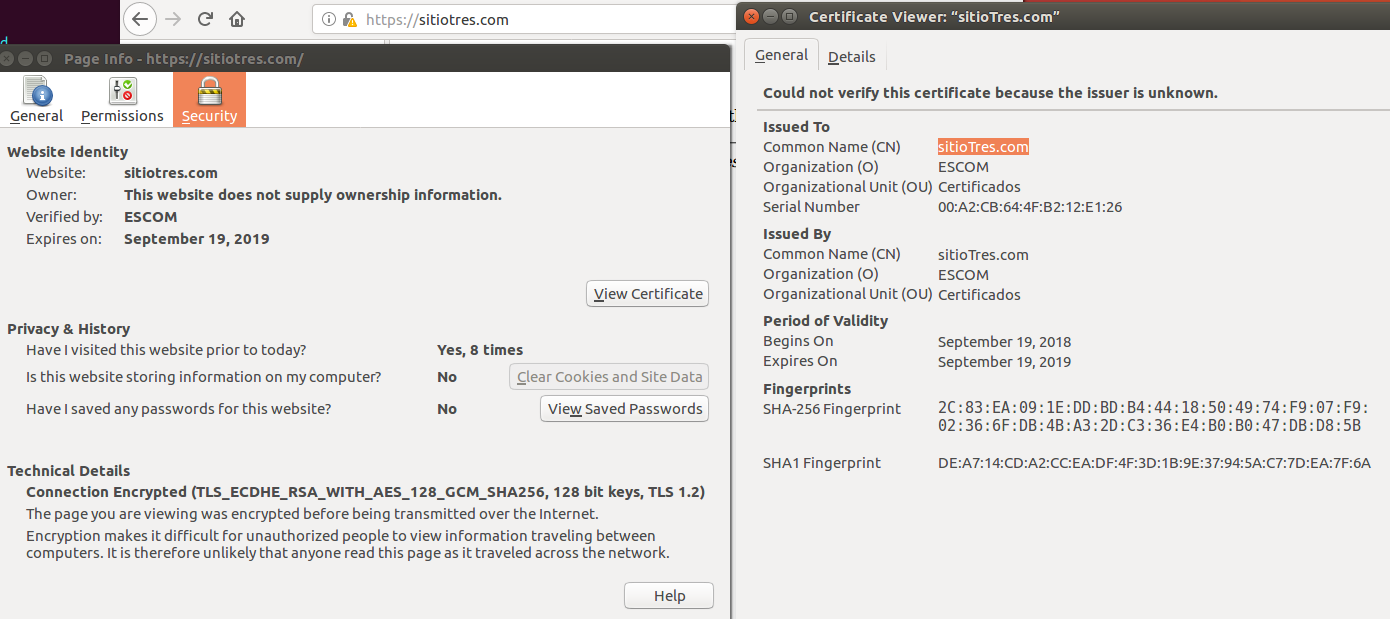


Nos avisará que nuestra conexión no es segura, para aceptar el certificado que acabamos de crear, seleccionamos el botón “Advanced” y después “Aadd Exception”





Confirmamos la excepción de seguridad y listo, ahora podemos ver que todo lo que hagamos dentro del sitio estará cifrado.



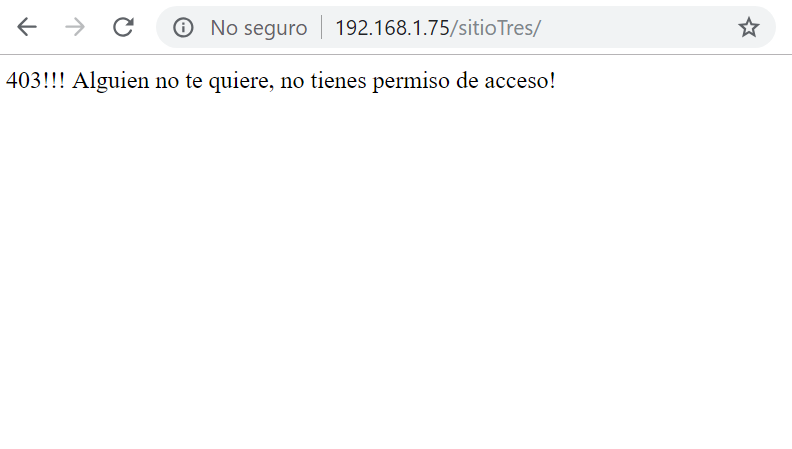
Los certificados proporcionados por terceros garantizan algunas características a la hora de tener conexión en una red, brindan un alto nivel de seguridad. Los certificados de terceros tienen generalmente un costo para poder utilizarlos, aunque existen patrocinios que dadas ciertas condiciones te permiten utilizar sus certificados de manera gratuita.

1. **Personalización de páginas de error para todos los sitios**
   1. Configurar al menos 3 Diferentes tipos de errores del sitio

Los html de los errores configurados deben estar en la carpeta /var/www/html, lo siguiente es modificar el archivo de configuración para agregar los errores personalizados, para ello agregamos la siguiente línea en el archivo conf por default /etc/apache2/sities-available/000-default.conf :

ErrorDocument xxx /nombreArchivoError.html

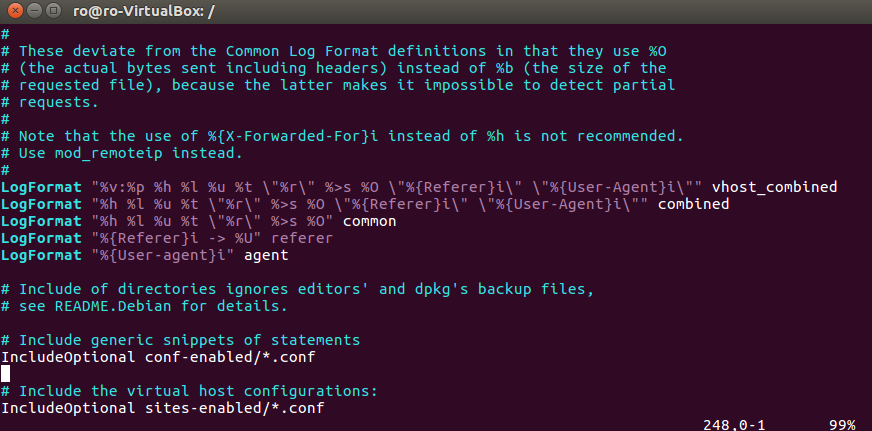




1. **Configuración de archivos de bitácoras y mensajes de error.**

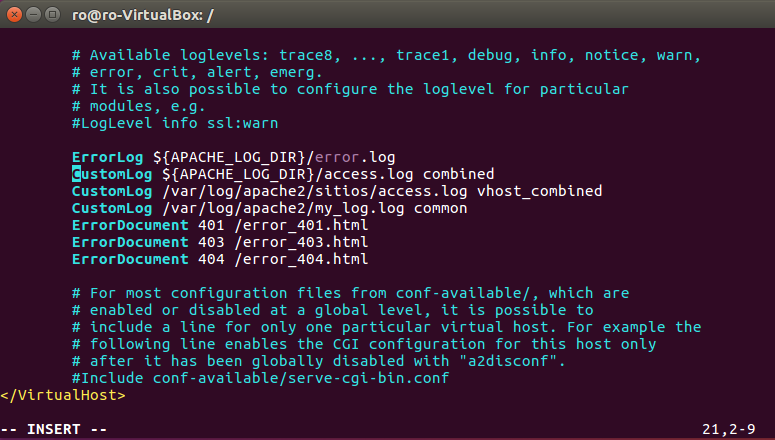
Apache tiene varias opciones de bandera que podemos utilizar para obtener información que nos interese ver como administradores, dicha configuración se realiza en el archivo /etc/apache2/apache2.con:

#vim /etc/apache2/apache2.conf #LogFormat banderas formatoSeleccionado



Hecho esto, procedemos a indicar en el archivo .conf el nuevo formato de log:

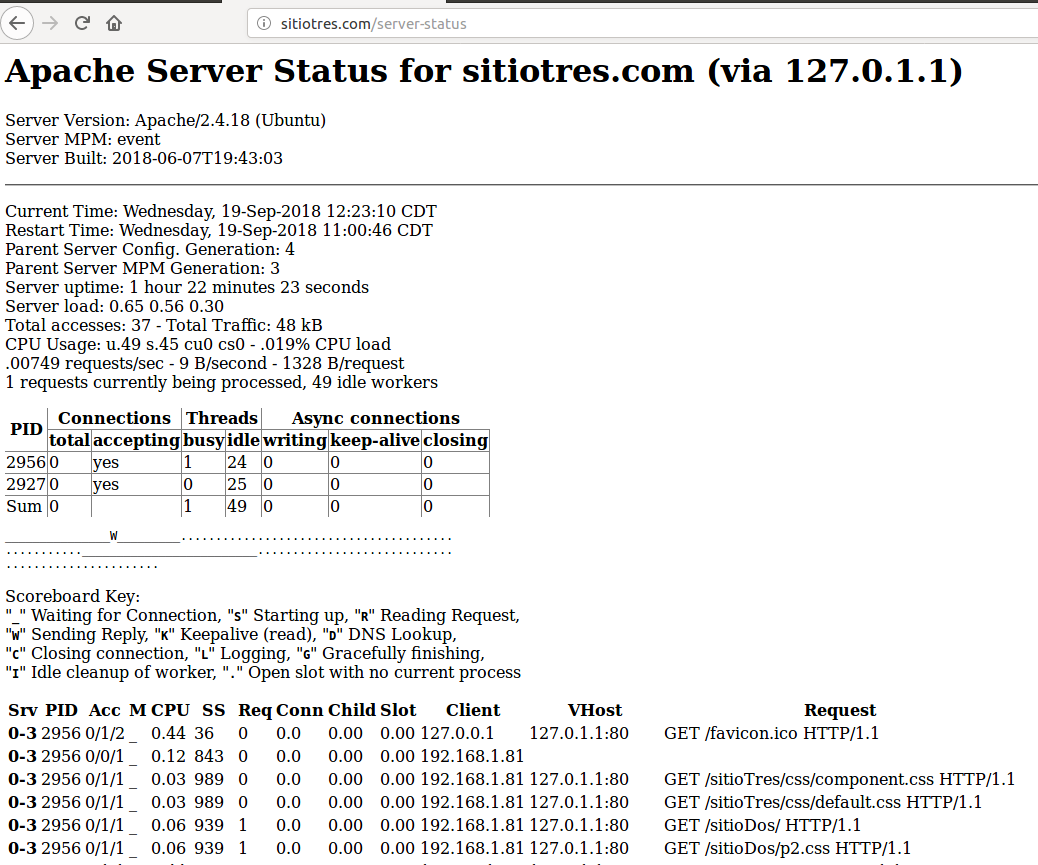
CustomLog rutaParaGuardarLog formatoSeleciconado

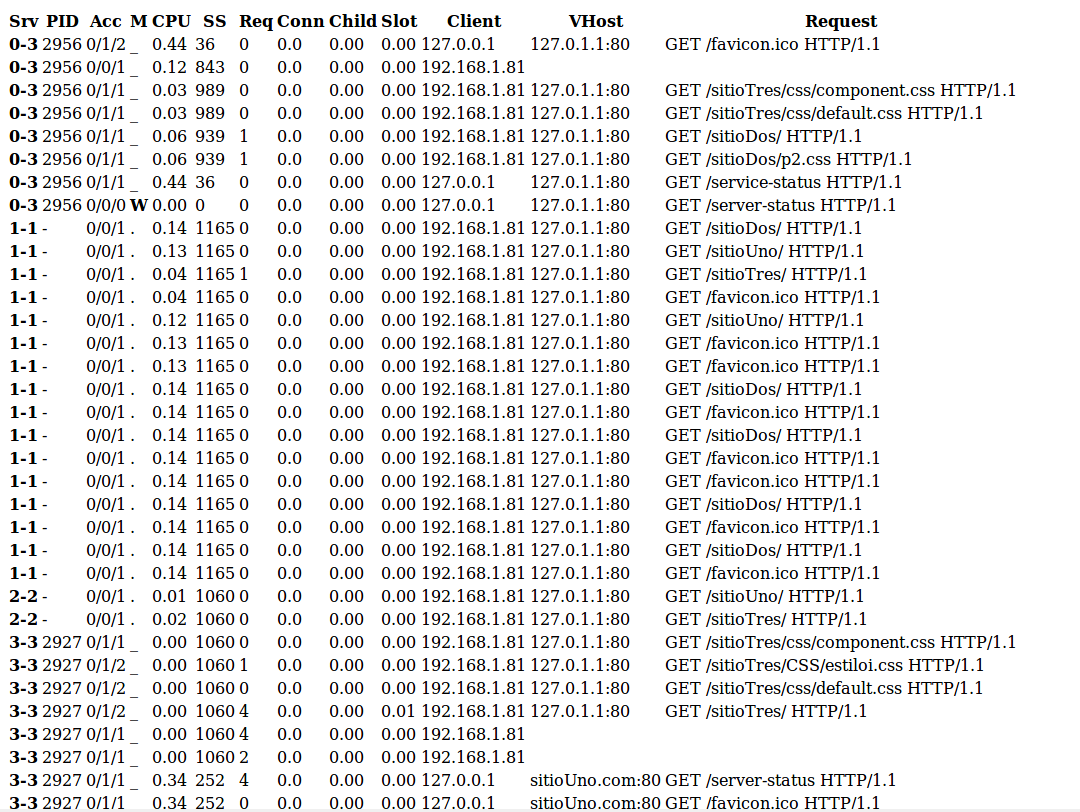


#service apahce2 reload

1. **Resumen de operación de forma dinámica (Sitios, solicitudes, estado del sistema y recursos consumidos)**
   1. Mostrar resumen de conexiones
   2. Mostrar resumen de consumo de recursos (Memoria/procesador/tiempo de ejecución)

Apache tiene una herramienta para mostrar el estado del servidor para cada sitio, podemos acceder a este recurso simplemente abriendo una nueva ventana de navegación y escribiendo en en la barra de direcciones nombredeSitio/server-status:





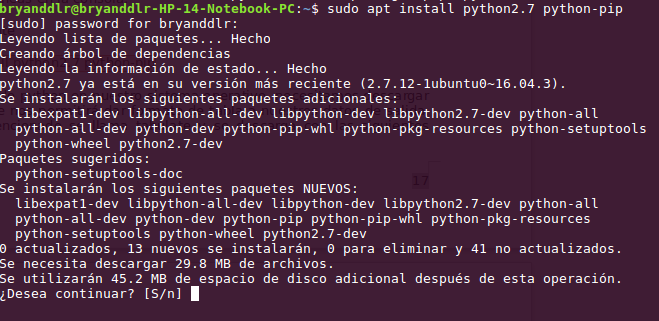
1. **Resumen de operación y de error del servidor**

Para la generación de resumenes se utilizó el lenguaje de programación python en su versión 2.7. Se eligió este lenguaje debido a la sencillez en que se puede realizar el manejo de ficheros (en este caso, los archivos log), además de que cuenta con diversos paquetes que permiten dar un formato presentable a los datos de salida.

Para instalar python en nuestro sistema operativo Ubuntu necesitamos ejecutar las siguientes líneas en la consola:

$sudo apt update

$sudo apt install python2.7 python-pip



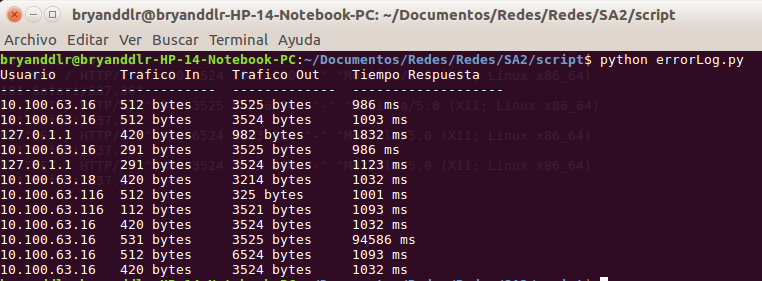
Una vez teniendo python en nuestro sistema operativo, necesitamos descargar un paquete que nos permitirá dar formato de tabla a nuestros datos de salida. El paquete mencionado se llama tabulate y se descarga con las siguientes líneas:

$pip install tabulate

Una vez instalado todo, ejecutamos los scripts.

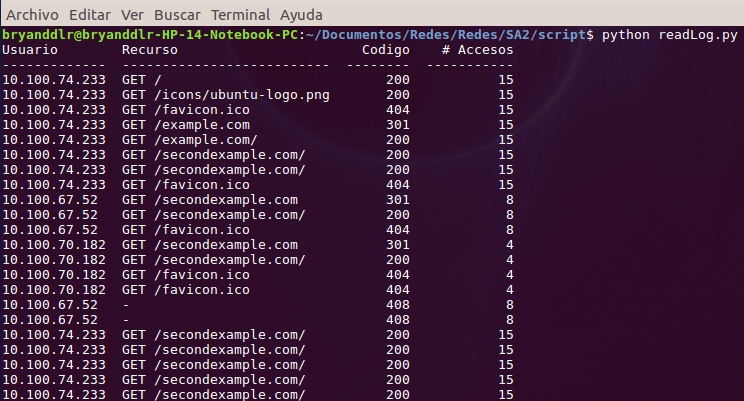
Para el resumen de operación ejecutamos el script opLog.py con las líneas:

$python opLog.py



Para el resumen de errores ejecutamos el script opLog.py con las líneas:

$python errorLog.py



**Referencias**

[1] The Apache HTTP Server Project, *Apache HTTP Server Version 2.4  Documentation,* [*https://httpd.apache.org/docs/2.4/*](https://httpd.apache.org/docs/2.4/)

[2] THe Apache HTTP Server Project, *Apache Virtual Hosts Documentation,* [*http://httpd.apache.org/docs/2.4/vhosts/*](http://httpd.apache.org/docs/2.4/vhosts/)

[3] COMODO Instant SSL, What is HTTPs?,[*https://www.instantssl.com/ssl-certificate-products/https.html*](https://www.instantssl.com/ssl-certificate-products/https.html)