antes de - son comandos utilizables

las palabras en rojo son estructuras de carpetas

las palabras en verde son nombres modificables

# Paso 1: Instalar Apache

Apache está disponible en los repositorios de software predeterminados de Ubuntu, lo que permite instalarlo con las herramientas convencionales de administración de paquetes.

Comencemos actualizando el índice de paquetes locales para que reflejen los últimos cambios anteriores:

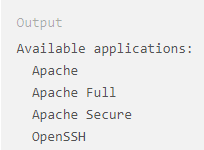
-sudo apt update

-sudo apt install apache2

# Paso 2: Ajustar el firewall

-sudo ufw app list

Obtendrá una lista de los perfiles de aplicación:



Como lo indica el resultado, hay tres perfiles disponibles para Apache:

Apache: este perfil abre solo el puerto 80 (tráfico web normal no cifrado)

Apache Full: este perfil abre el puerto 80 (tráfico web normal no cifrado) y el puerto 443 (tráfico TLS/SSL cifrado)

Apache Secure: este perfil abre solo el puerto 443 (tráfico TLS/SSL cifrado)

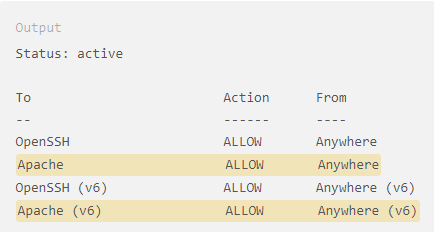
Se recomienda habilitar el perfil más restrictivo, que de todos modos permitirá el tráfico que se configuró.

-sudo ufw allow 'Apache'

Puede verificar el cambio escribiendo lo siguiente:

-sudo ufw status

El resultado proporcionará una lista del tráfico de HTTP que se permite:



Como lo indica el resultado, el perfil se activó para permitir el acceso al servidor web Apache.

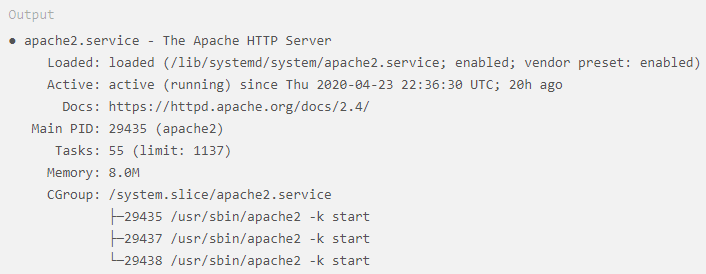
# Paso 3: Comprobar su servidor web

Al final del proceso de instalación, Ubuntu 20.04 inicia Apache. El servidor web ya debería estar activo.

Realice una verificación con el sistema init systemd para saber si se encuentra en ejecución el servicio escribiendo lo siguiente:

-sudo systemctl status apache2

el resultado es algo como esto



Puede acceder a la página de destino predeterminada de Apache para confirmar que el software funcione correctamente mediante su dirección IP: Si no conoce la dirección IP de su servidor, puede obtenerla de varias formas desde la línea de comandos.

Intente escribir esto en la línea de comandos de su servidor:

-hostname -I

Otra opción es utilizar la herramienta Icanhazip, que debería proporcionar su dirección IP pública como aparece en otra ubicación en Internet:

-curl -4 icanhazip.com

Cuando tenga la dirección IP de su servidor, introdúcela en la barra de direcciones de su navegador:

http://tu \_direccion\_ip

Debería ver la página web predeterminada de Apache en Ubuntu 20.04:



# Paso 4: Administrar el proceso de Apache

Ahora que el servidor web está listo y en funcionamiento, repasemos algunos comandos de administración básicos con systemctl.

Para detener su servidor web, escriba lo siguiente:

-sudo systemctl stop apache2

Para iniciar el servidor web cuando no esté activo, escriba lo siguiente:

-sudo systemctl start apache2

Para detener y luego iniciar el servicio de nuevo, escriba lo siguiente:

-sudo systemctl restart apache2

Si solo realiza cambios de configuración, Apache a menudo puede recargarse sin cerrar conexiones. Para hacerlo, utilice este comando:

-sudo systemctl reload apache2

Por defecto, Apache está configurado para iniciarse automáticamente cuando el servidor lo hace. Si no es lo que quiere, deshabilitar este comportamiento escribiendo lo siguiente:

-sudo systemctl disable apache2

Para volver a habilitar el servicio de modo que se cargue en el inicio, escriba lo siguiente:

-sudo systemctl enable apache2

Ahora, Apache debería iniciarse de forma automática cuando el servidor lo haga de nuevo.

# **Paso 5: Configurar hosts virtuales**

se hace para alojar varios sitios web

Ubuntu 20.04 tiene habilitado un bloque de servidor por defecto, que está configurado para proporcionar documentos del directorio /var/www/html. Si bien esto funciona bien para un solo sitio, puede ser difícil de manejar si aloja varios. En vez de modificar /var/www/html, vamos a crear una estructura de directorios dentro de /var/www para un sitio your\_domain y dejaremos /var/www/html como directorio predeterminado que se suministrará si una solicitud de cliente no coincide con otros sitios.

Cree el directorio para your\_domain de la siguiente manera:

-sudo mkdir /var/www/your\_domain

A continuación, asigne la propiedad del directorio con la variable de entorno $USER:

-sudo chown -R $USER:$USER /var/www/your\_domain

Los permisos de los roots web deberían ser correctos. Para asegurarse de que sus permisos sean correctos y permitir al propietario leer, escribir y ejecutar los archivos, y a la vez conceder solo permisos de lectura y ejecución a los grupos y terceros, puede ingresar el siguiente comando:

-sudo chmod -R 755 /var/www/your\_domain

A continuación, cree una página de ejemplo index.html utilizando nano o su editor favorito:

-sudo nano /var/www/your\_domain/index.html

Dentro de ella, agregue el siguiente ejemplo de HTML:

<html>

<head>

<title>Welcome to Your\_domain!</title>

</head>

<body>

<h1>Success! The your\_domain virtual host is working!</h1>

</body>

</html>

Guarde ctrl x y cierre el archivo cuando termine.

Para que Apache proporcione este contenido, es necesario crear un archivo de host virtual con las directivas correctas. En lugar de modificar el archivo de configuración predeterminado situado en /etc/apache2/sites-available/000-default.conf directamente, vamos a crear uno nuevo en /etc/apache2/sites-available/your\_domain.conf:

-sudo nano /etc/apache2/sites-available/your\_domain.conf

Péguelo en el siguiente bloque de configuración, similar al predeterminado, pero actualizado para nuestro nuevo directorio y nombre de dominio:

<VirtualHost \*:80>

ServerAdmin webmaster@localhost

ServerName your\_domain

ServerAlias www.your\_domain

DocumentRoot /var/www/your\_domain

ErrorLog ${APACHE\_LOG\_DIR}/error.log

CustomLog ${APACHE\_LOG\_DIR}/access.log combined

</VirtualHost>

Tenga en cuenta que cambiamos DocumentRoot por nuestro nuevo directorio y ServerAdmin por un correo electrónico al que pueda acceder el administrador del sitio your\_domain. También agregamos dos directivas: ServerName, que establece el dominio de base que debería coincidir para esta definición de host virtual, y ServerAlias, que define más nombres que deberían coincidir como si fuesen el nombre de base.

Guarde y cierre el archivo cuando termine.

Habilitaremos el archivo con la herramienta a2ensite:

-sudo a2ensite your\_domain.conf

Deshabilite el sitio predeterminado definido en 000-default.conf:

-sudo a2dissite 000-default.conf

A continuación, realizaremos una prueba para ver que no haya errores de configuración:

-sudo apache2ctl configtest

Debería obtener el siguiente resultado:

Output

Syntax OK

Reinicie Apache para implementar sus cambios:

* systemctl reload apache2

Con esto, Apache debería ser el servidor de su nombre de dominio. Puede probarlo visitando http://your\_domain, donde debería ver algo como esto:



# Paso 6: Familiarizarse con archivos y direcciones importantes de Apache

Ahora que sabe administrar el propio servicio de Apache, debe tomarse unos minutos para familiarizarse con algunos directorios y archivos importantes.

### Contenido

* /var/www/html: el contenido web real, que por defecto solo consta de la página predeterminada de Apache que vio antes, se proporciona desde el directorio /var/www/html. Esto se puede cambiar modificando los archivos de configuración de Apache.

### Configuración del servidor

* /etc/apache2: el directorio de configuración de Apache. En él se encuentran todos los archivos de configuración de Apache.
* /etc/apache2/apache2.conf: el archivo principal de configuración de Apache. Esto se puede modificar para realizar cambios en la configuración general de Apache. Este archivo administra la carga de muchos de los demás archivos del directorio de configuración.
* /etc/apache2/ports.conf: este archivo especifica los puertos en los que Apache escuchará. Por defecto, Apache escucha en el puerto 80. De forma adicional, lo hace en el 443 cuando se habilita un módulo que proporciona capacidades SSL.
* /etc/apache2/sites-available/: el directorio en el que se pueden almacenar hosts por sitio. Apache no utilizará los archivos de configuración de este directorio a menos que estén vinculados al directorio sites-enabled. Normalmente, toda la configuración de bloques de servidor se realiza en este directorio y luego se habilita al vincularse al otro directorio con el comando a2ensite.
* /etc/apache2/sites-enabled/: el directorio donde se almacenan hosts virtuales por sitio habilitados. Normalmente, se crean vinculando los archivos de configuración del directorio sites-available con a2ensite. Apache lee los archivos de configuración y los enlaces de este directorio cuando se inicia o se vuelve a cargar para compilar una configuración completa.
* /etc/apache2/conf-available/ y /etc/apache2/conf-enabled/: estos directorios tienen la misma relación que los directorios sites-available y sites-enabled, pero se utilizan para almacenar fragmentos de configuración que no pertenecen a un host virtual. Los archivos del directorio conf-available pueden habilitarse con el comando -a2enconf y deshabilitarse con el comando -a2disconf.
* /etc/apache2/mods-available/ y /etc/apache2/mods-enabled/: estos directorios contienen los módulos disponibles y habilitados, respectivamente. Los archivos que terminan en .load contienen fragmentos para cargar módulos específicos, mientras que los archivos que terminan en .conf contienen la configuración para esos módulos. Los módulos pueden habilitarse y deshabilitarse con los comandos a2enmod y a2dismod.

### Registros del servidor

* /var/log/apache2/access.log: por defecto, cada solicitud enviada a su servidor web se asienta en este archivo de registro a menos que Apache esté configurado para no hacerlo.
* /var/log/apache2/error.log: por defecto, todos los errores se registran en este archivo. La directiva LogLevel de la configuración de Apache especifica el nivel de detalle de los registros de error.