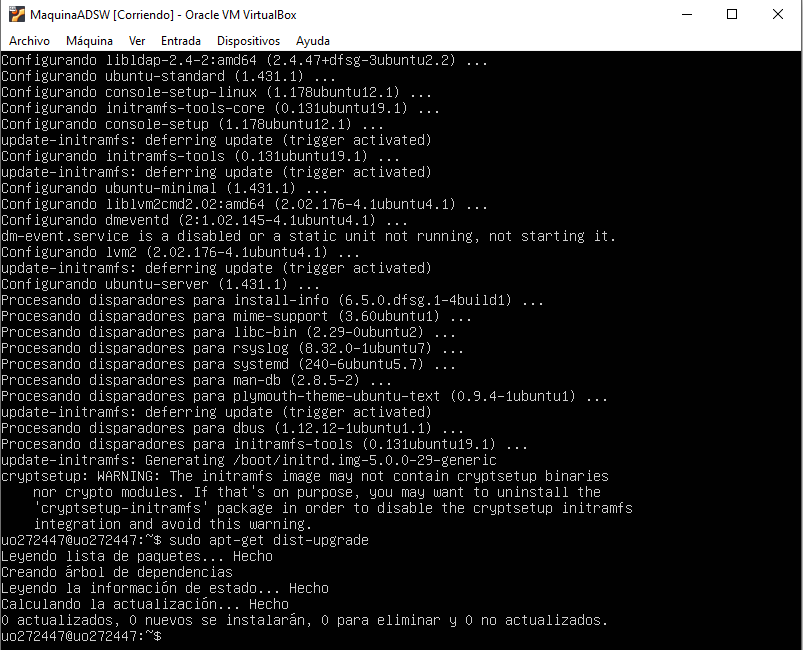
**Arranque máquina en vBox fallaba (VMSetError: Call to WHvSetupPartition failed: ERROR\_SUCCESS (Last=0xc000000d/87)), solución:**

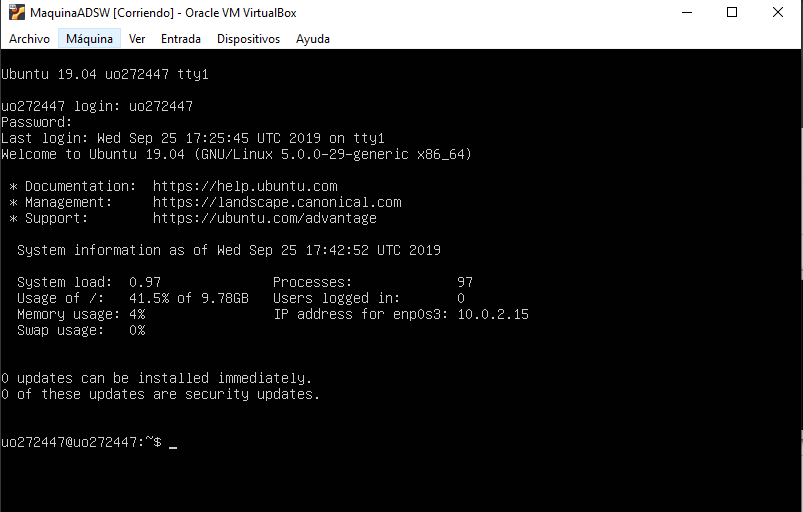
RUN > CMD > bcdedit /set hypervisorlaunchtype off

then reboot host machine

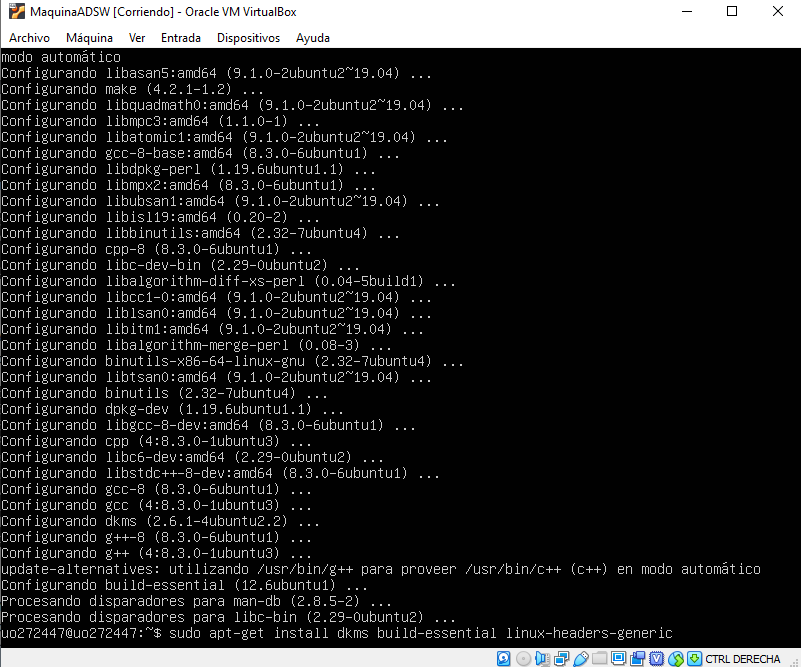
UBUNTU

|  |  |
| --- | --- |
| **1.1** | *Instalar una máquina Ubuntu Server LTS virtualizada y actualizada* |

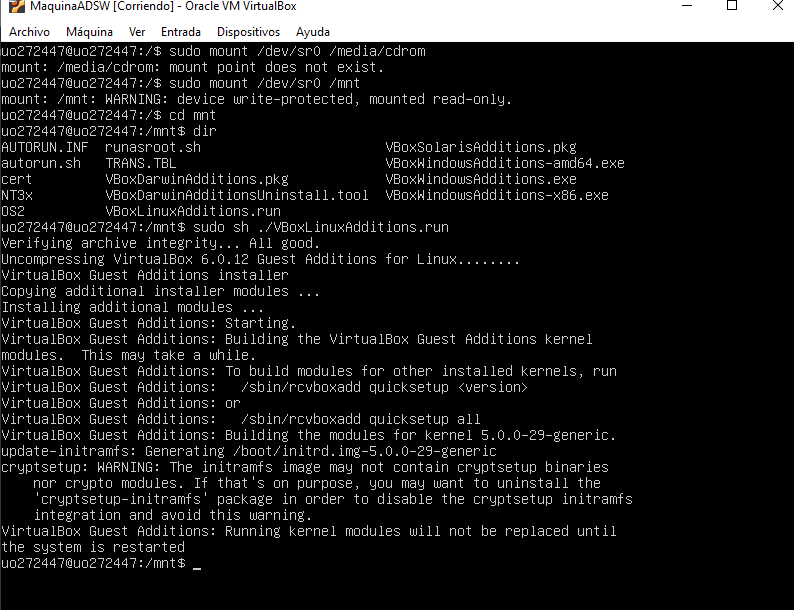




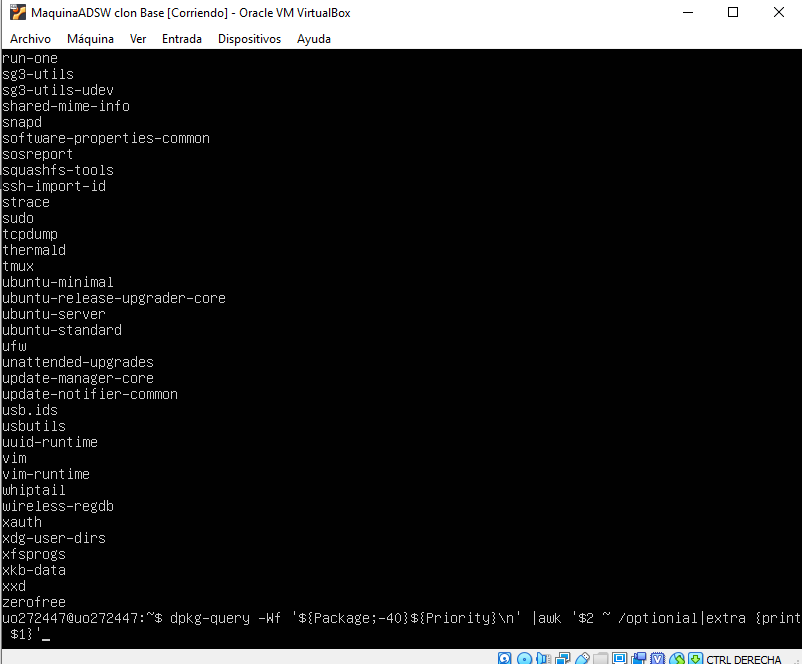
|  |  |
| --- | --- |
| **1.2** | *Optimizar la virtualización de la máquina virtual instalada* |



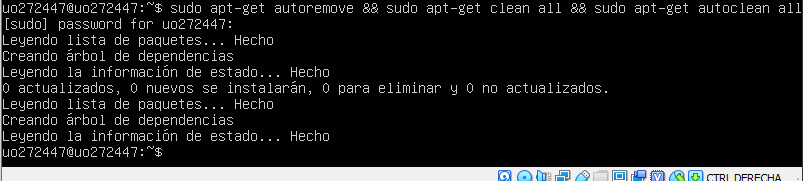
Al ser la versión 19 se utiliza la ruta /mnt en lugar de /media/cdrom



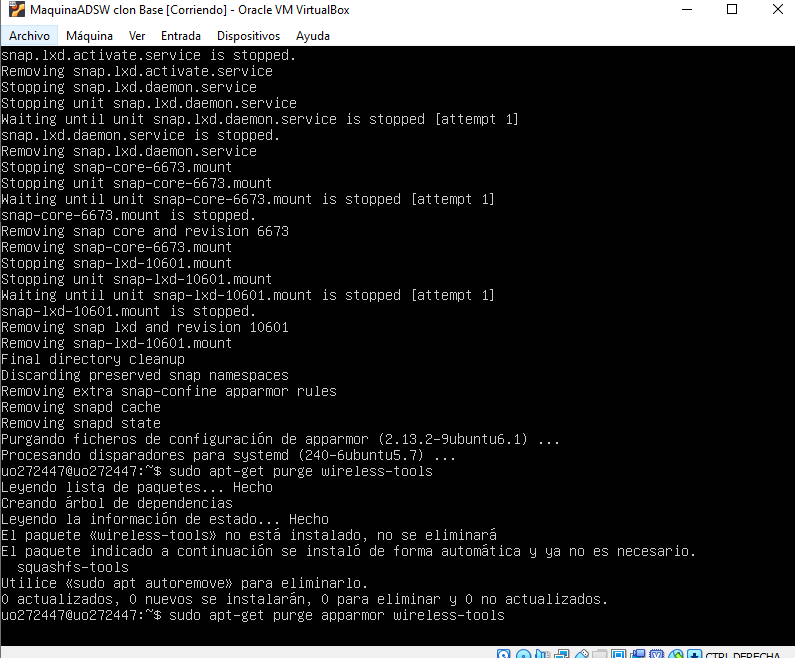
|  |  |
| --- | --- |
| **1.3** | *Optimizar la instalación de paquetes y de usuarios de la máquina* |



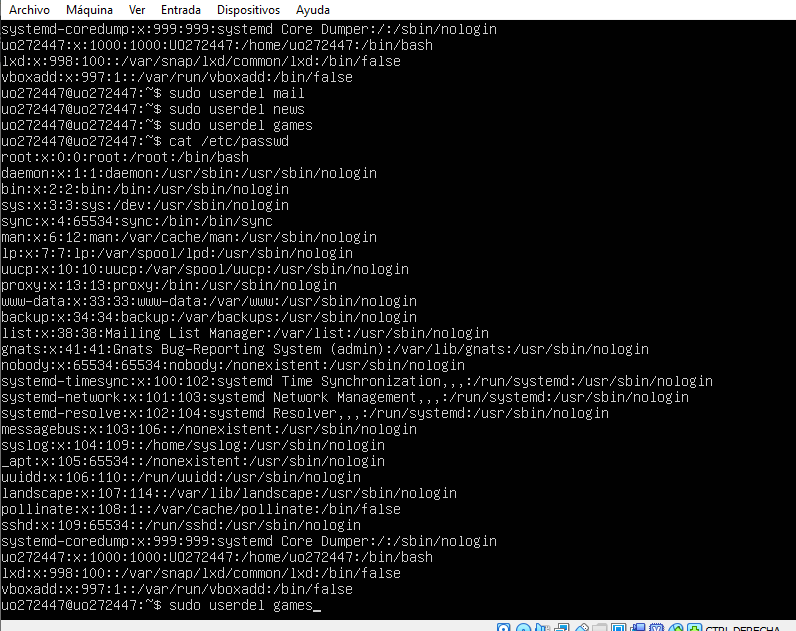
Limpieza de paquetes



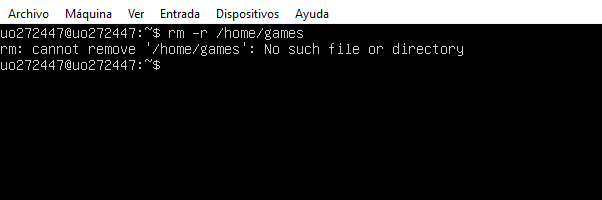
Por el momento se eliminan los mismos paquetes que los del ejemplo.



Se eliminan los mismos usuarios del ejemplo, irc, mail, news y games

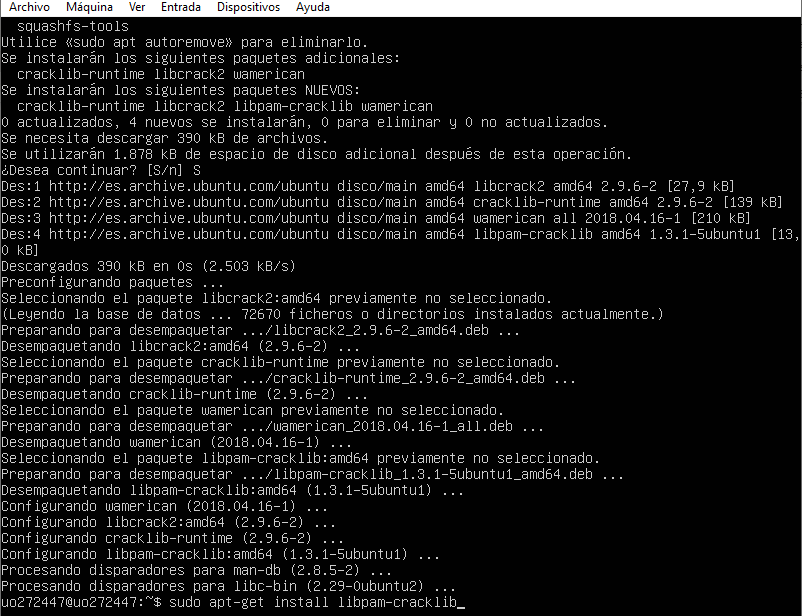


Se ha comprobado a eliminar el directorio home de cada uno por si acaso

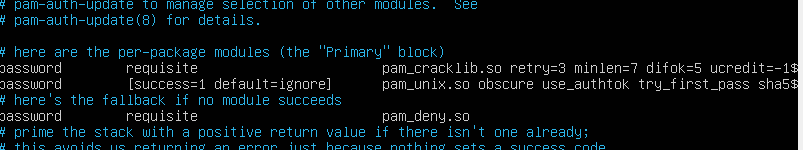


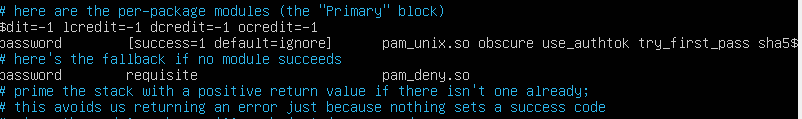
|  |  |
| --- | --- |
| **1.4** | *Prohibir el uso de passwords comunes* |

Instalación librería

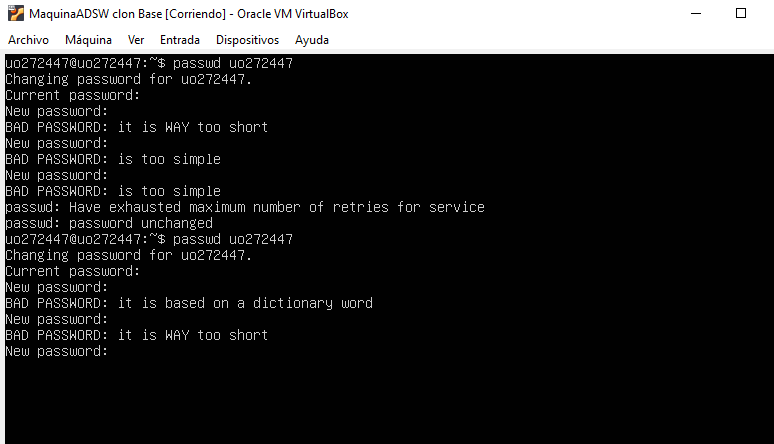


Se cambia los requisitos de las contraseñas, como mi contrseña tiene una longitud de 7 he usado minlen = 7 para no provocar problemas.



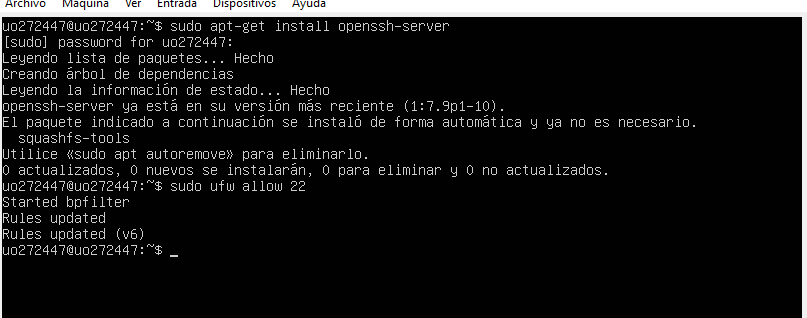


Se comprueba que no admiten las contraseñas: a – aaabbbcccdddee – crem123 - butterfly123 – “”



|  |  |
| --- | --- |
| **1.5** | *Habilitar el acceso vía ssh automático a una máquina virtual* |

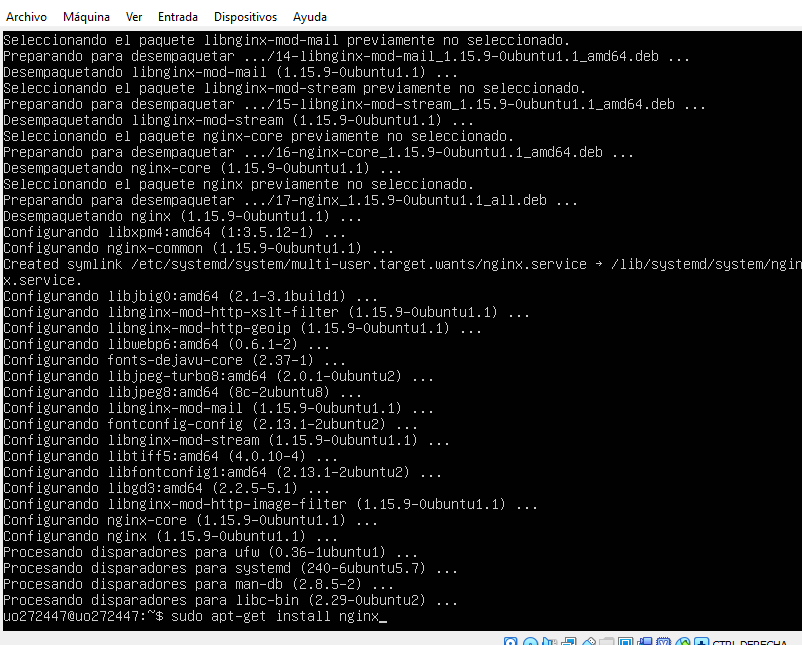
Instalación ssh hecha por defecto en la instalación del servidor, se dan permisos:



NGINX

|  |  |
| --- | --- |
| **1.1** | *Instalar Nginx bajo Ubuntu Server y comprobar que funciona* |

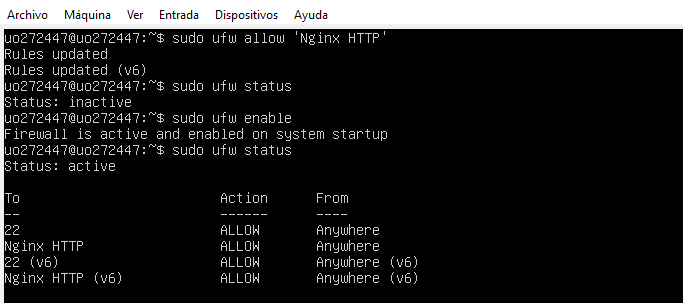
Instalación



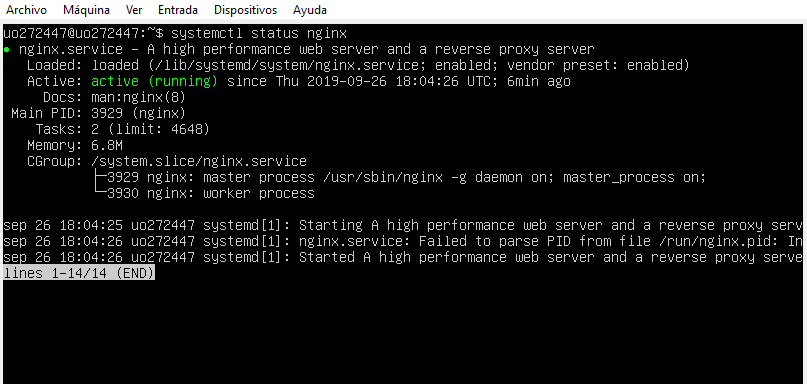
Comprobar que se ha registrado



Comprobar que esta bien configurado



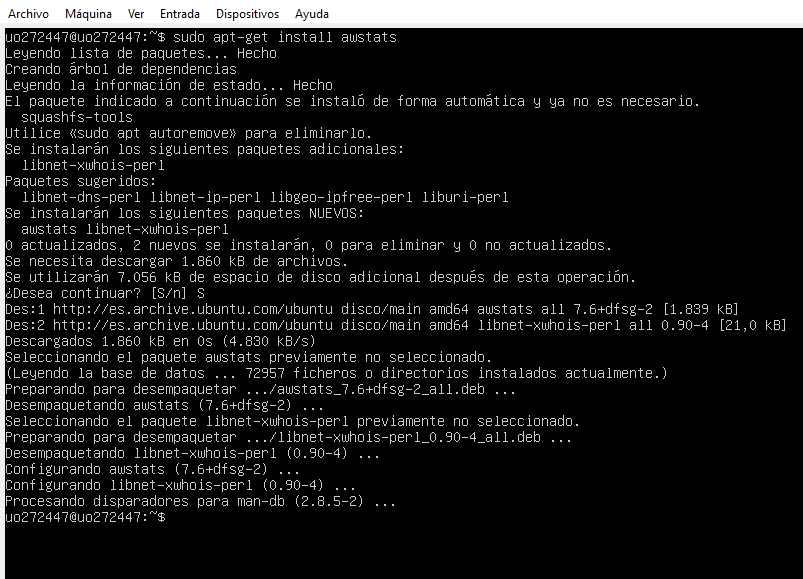
Ver que esta funcionando



|  |  |
| --- | --- |
| **1.2** | *Mediante el uso de herramientas de análisis y consulta de las webs adecuadas, analizar y describir los potenciales problemas de seguridad que podría tener la versión de Nginx instalada en el servidor Ubuntu Server* |

|  |  |
| --- | --- |
| **1.4** | *Ejecutar en análisis de ficheros de log con la herramienta awstats* |

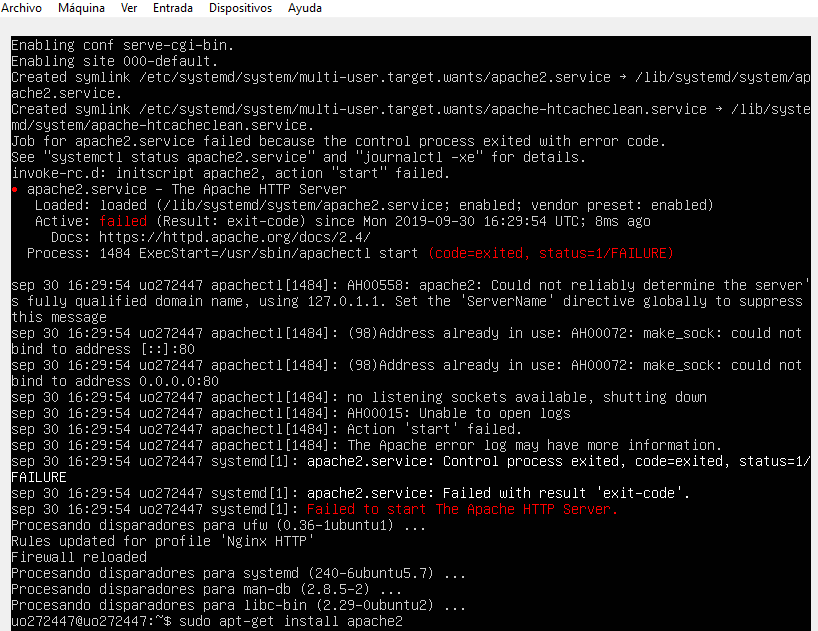
Instalacion



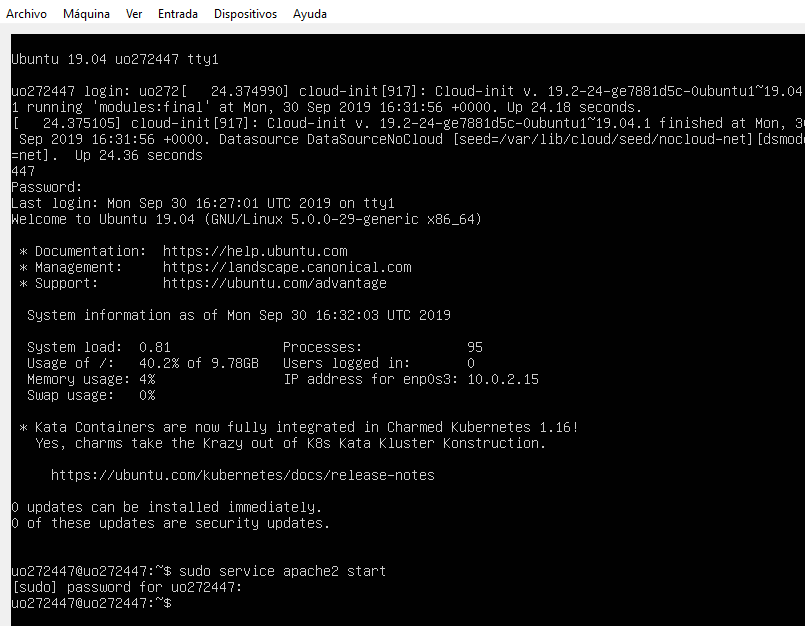
APACHE

|  |  |
| --- | --- |
| **1.1** | *Instalar Apache bajo Ubuntu Server y comprobar que funciona* |

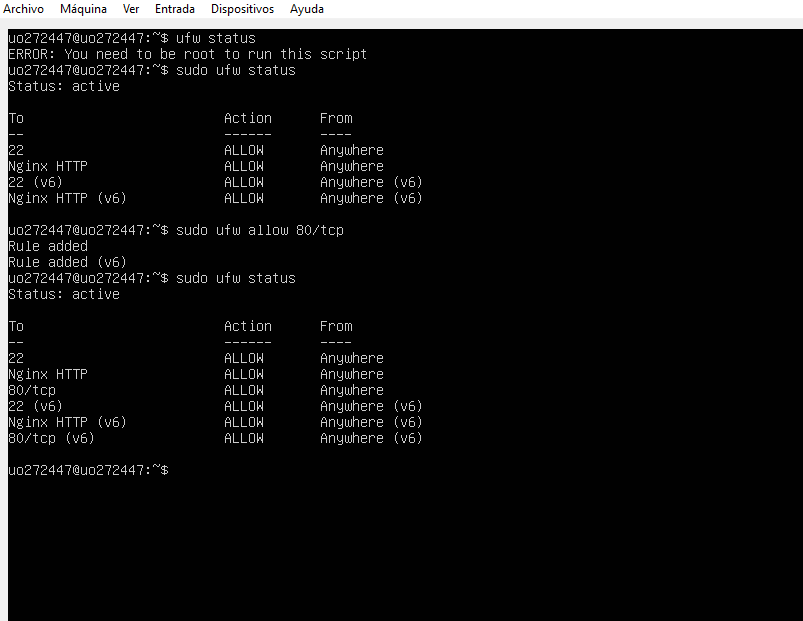
Instalación



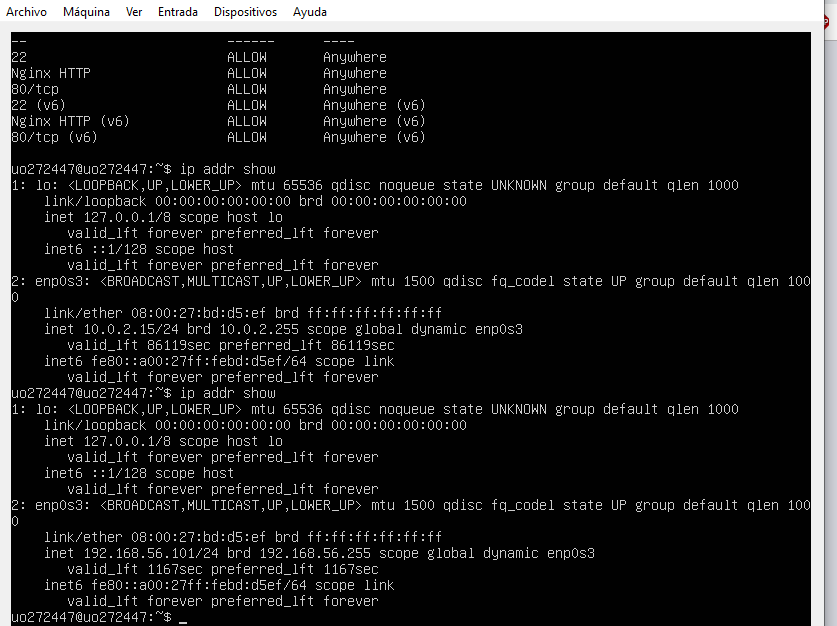
Al dar error al arrancar, se reinicia la maquina y se fuerza el arranque

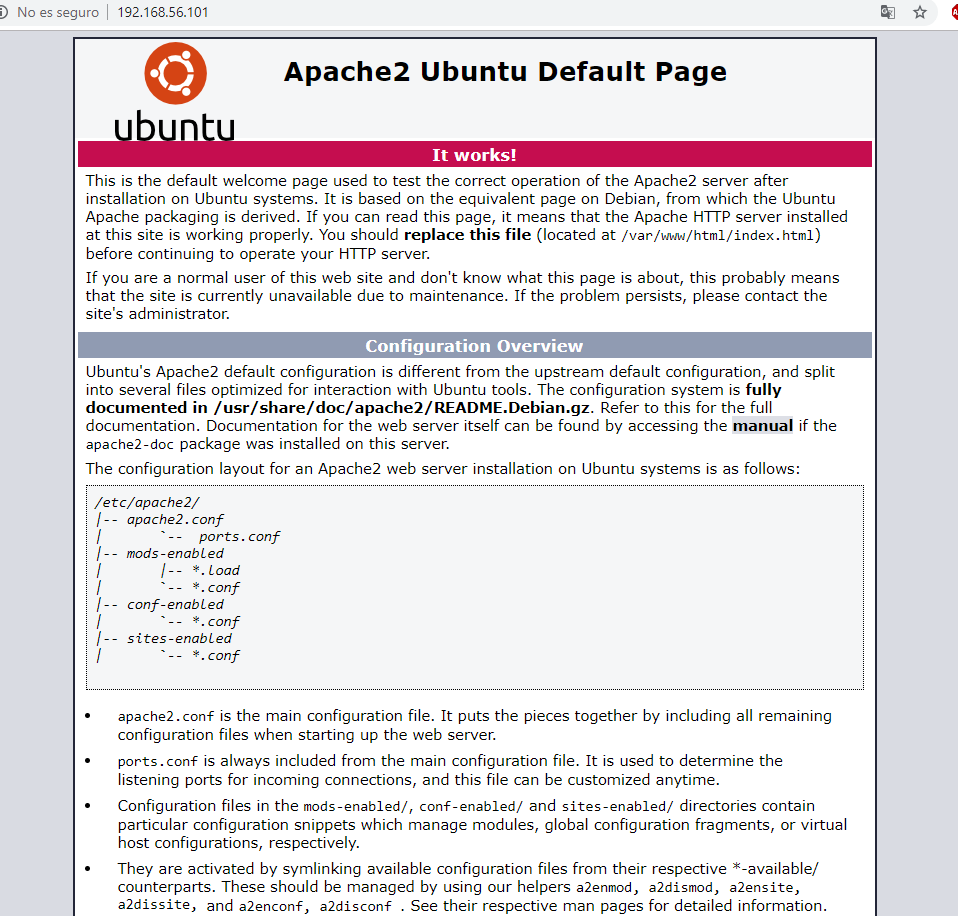


Se comprueba que se puedan recibir peticiones por el puerto 80, como esta cerrado se abre.



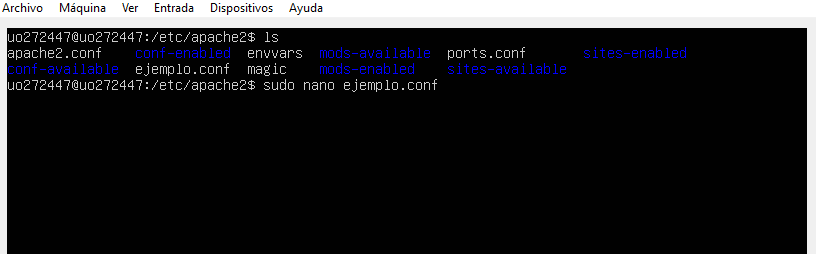
Comprobar dirección ip asignada para acceder al servidor, en este caso [192.168.56.101](http://192.168.56.101/)



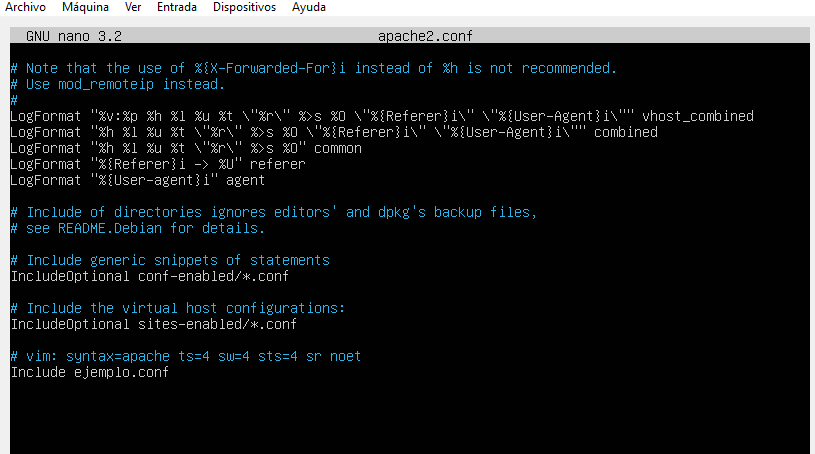


|  |  |
| --- | --- |
| **1.2** | *Crear un fichero de configuración de forma que se puedan hacer en él todos los cambios de configuración que puedan hacerse en apache2.conf* |

Se crea un fichero ejemplo.conf en la misma carpeta donde se encuentra apache2.conf

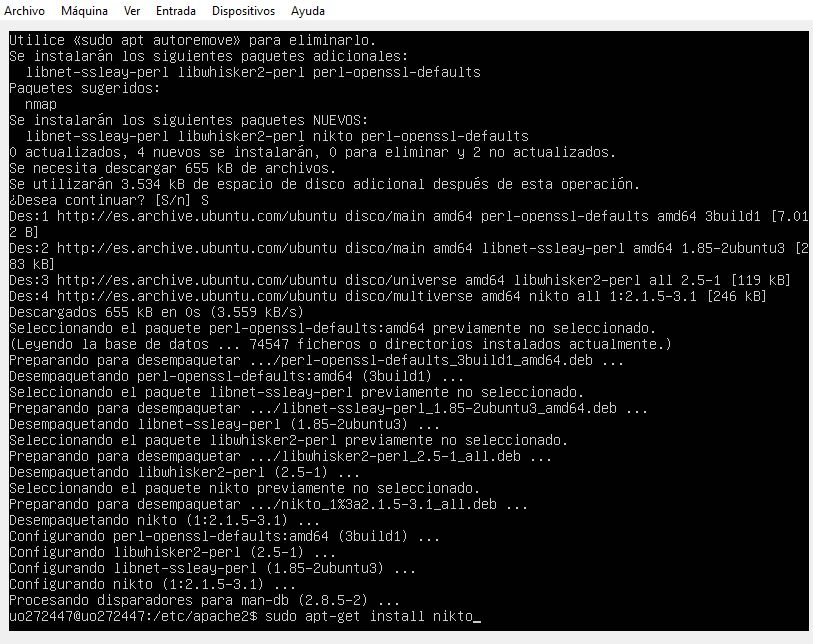


Se incluye el archivo en apache2.conf

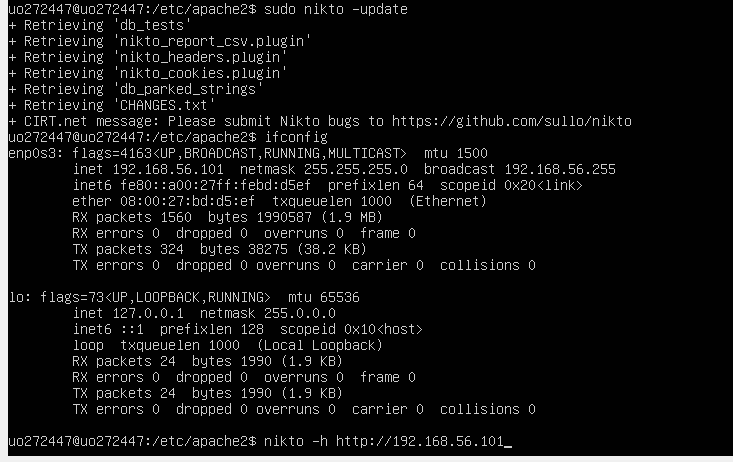


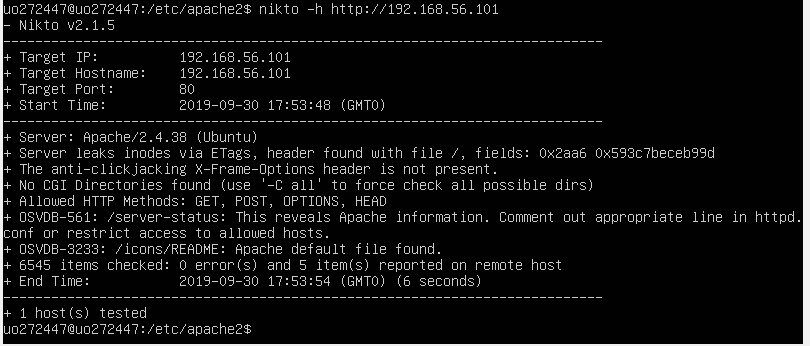
|  |  |
| --- | --- |
| **1.3** | *Mediante el uso de herramientas de análisis y consulta de las webs adecuadas, analizar y describir los potenciales problemas de seguridad que podría tener la versión de Apache instalada en el servidor Ubuntu Server* |

Se instala la herramienta nikto para ver las vulnerabilidades del servidor



Se actualiza, se comprueba la ip asignada y se lanza con la opción -h



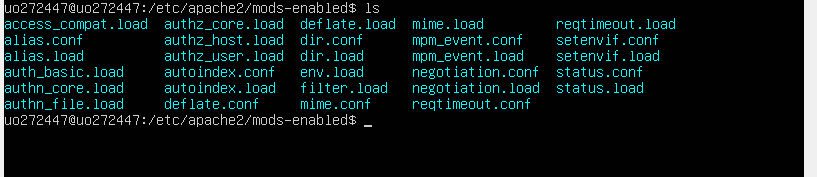


Se puede ver que dirección IP utiliza 192.168.56.101 y el puerto 80, la versión de servidor Apache 2.4.38 y que está corriendo sobre un sistema operativo Ubuntu.

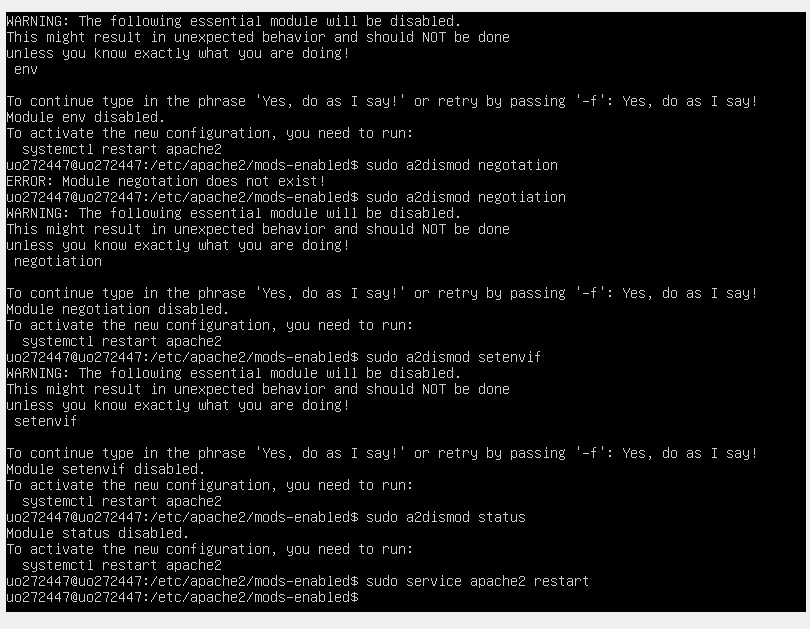
Sabiendo la versión del servidor, sería suficiente con visitar la página de apache para ver el listado de vulnerabilidades que tiene ese servidor: <https://httpd.apache.org/security/vulnerabilities_24.html>

|  |  |
| --- | --- |
| **1.4** | *Hacer que Apache tenga el menor nº de módulos posible para dar un servicio básico consistente en servir páginas web estáticas* |

Módulos instalados inicialmente



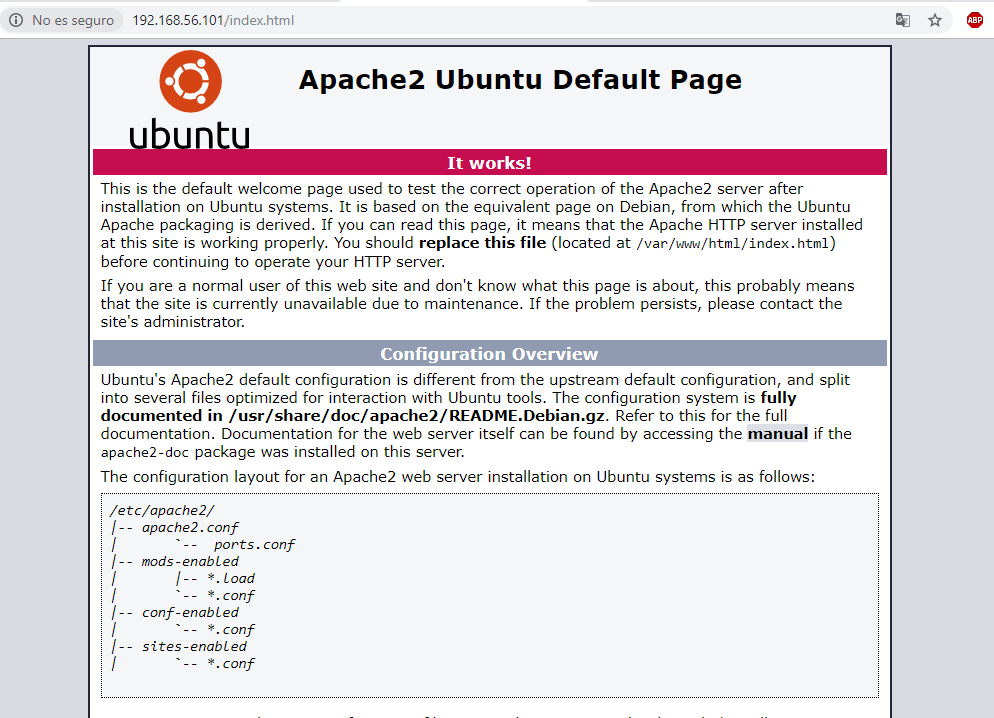
Se procede a desinstalar los módulos para dejar los mínimos posibles para dar un servicio básico y se reinicia el servicio.



Comprobamos que se han eliminado

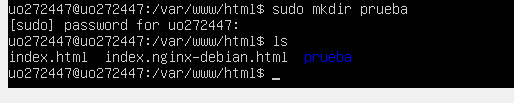


Comprobamos que sigue funcionando

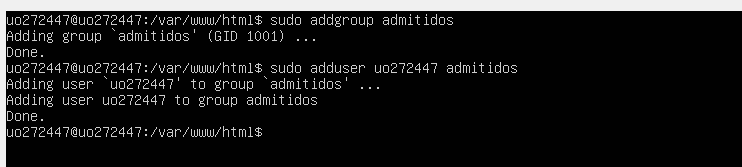


|  |  |
| --- | --- |
| **1.5** | *Hacer que todos los contenidos que dependan de un directorio llamado como vuestro nombre en el servidor puedan ser vistos por todo el mundo, pero solo un usuario creado ex-profeso (cuyo nombre es vuestro UO) pueda editarlos y/o añadir nuevos* |

Se crea el directorio prueba en la ruta var/www/html



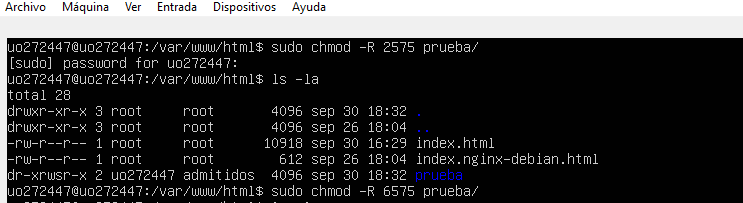
Se crea el grupo admitidos, (los usuarios pertenecientes a este grupo podrán añadir o crear nuevos), y añadimos el usuario uo272447 al grupo



Hacemos al usuario y grupo dueños de la carpeta prueba

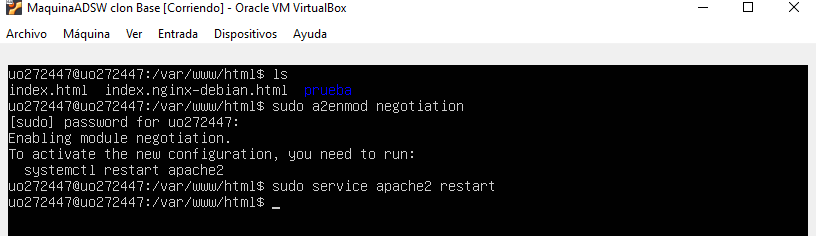


Damos permiso a los miembros del grupo para que solo ellos puedan editar y añadir nuevos archivos activando setgid y setuid.

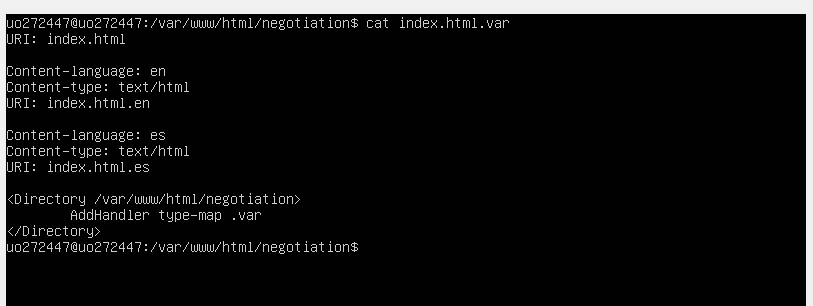


|  |  |
| --- | --- |
| **2.1** | *Mediante el uso de mod\_negotiation, crear un sistema de internacionalización de páginas web* |

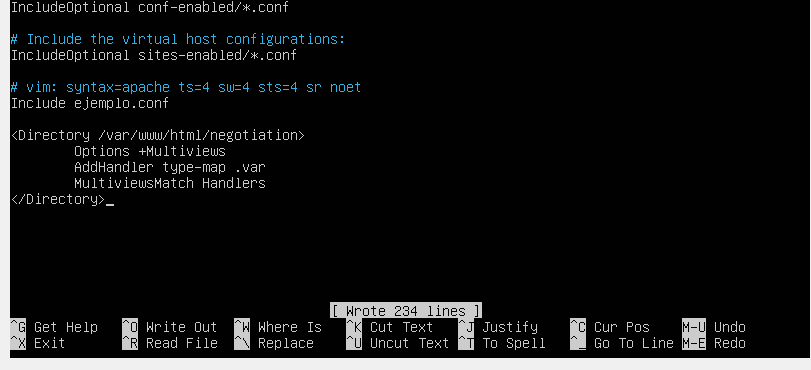
Asegurarse de que mod\_negotiation está activo:



Indicamos los archivos que se tienen que servir en función del idioma, si es ingles o español

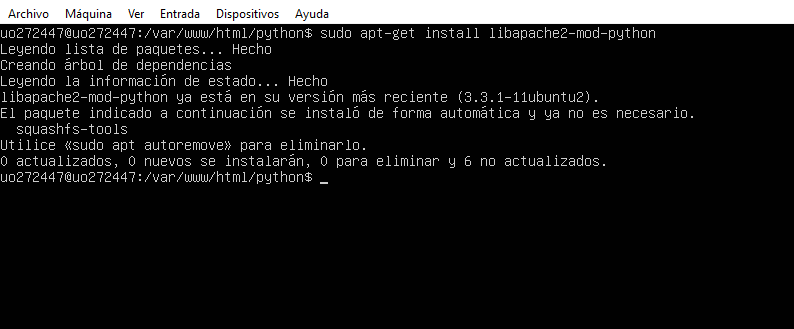


Se añade una directiva en apache2.conf para que multiviews para que decida cual es la versión mas adecuada que tiene que servir

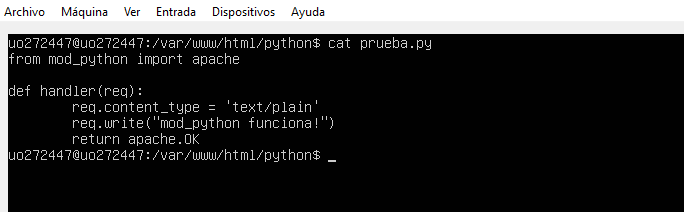


|  |  |
| --- | --- |
| **2.2** | *Instalar un handler Python para servir contenidos web mediante archivos .py* |

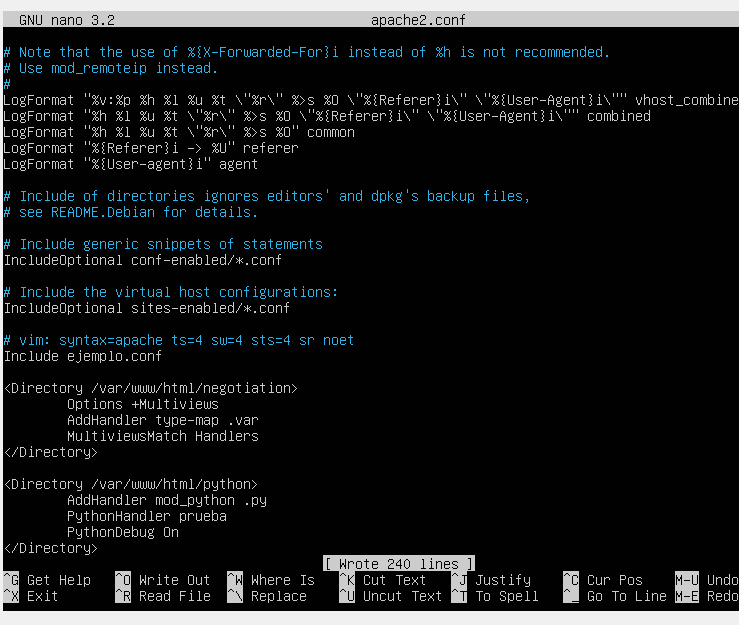
Se instala el mod\_python



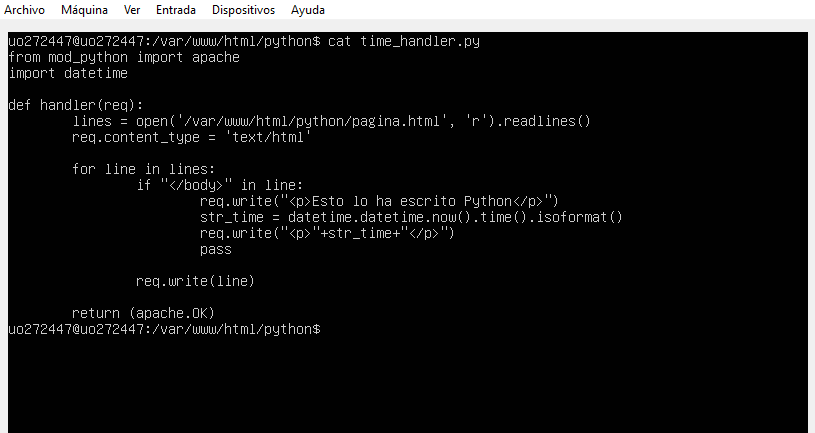
Se crea un pequeño programa para comprobar que el mod funciona



Finalmente se añade la directiva en apache2.conf para que cualquier petición .py sea procesada por el mod



Para darle mas seguridad se utilizará el fichero time\_handler.py con el siguiente código

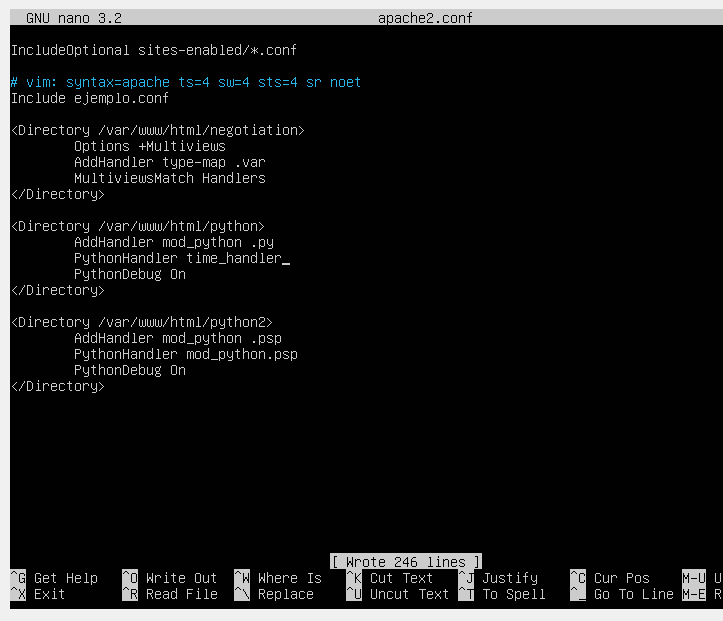


Y se modifica el apache2.conf para utilizar este ultimo archivo en lugar de prueba.py



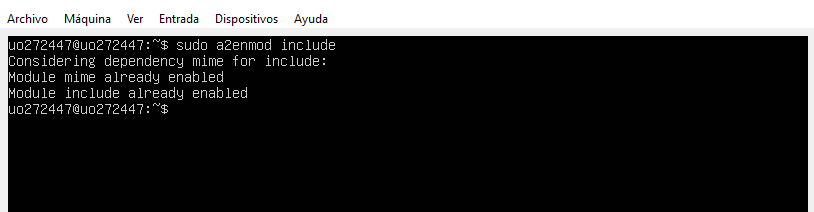
|  |  |
| --- | --- |
| **2.3** | *Habilitar soporte para páginas .psp* |

Basta con añadir, como en el caso anterior, en el fichero apache2.conf para que coja cualquier directiva .psp

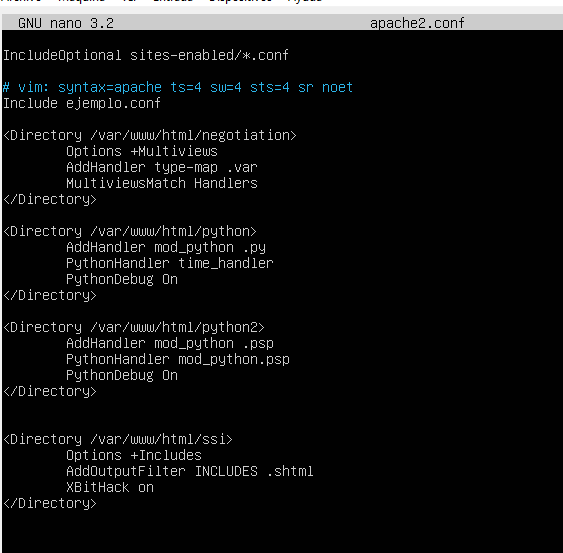


|  |  |
| --- | --- |
| **2.4** | *Habilitar SSI en un directorio o host virtual particular* |

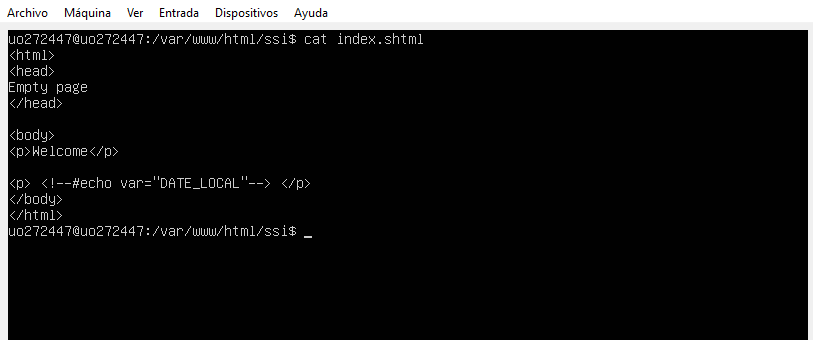
habilitar el módulo de Apache que permite el uso de SSI (mod\_include), pero ya existe



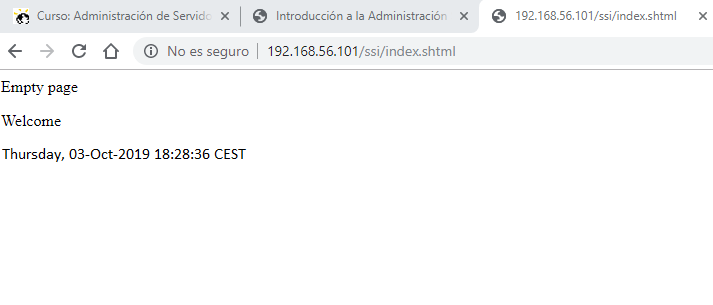
indicamos que vamos a permitir SSI sobre los ficheros de un directorio concret



Se crea un archivo shtml que utiliza ssi

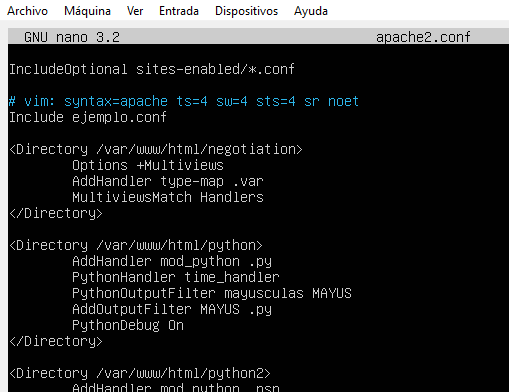


Comprobamos que funciona

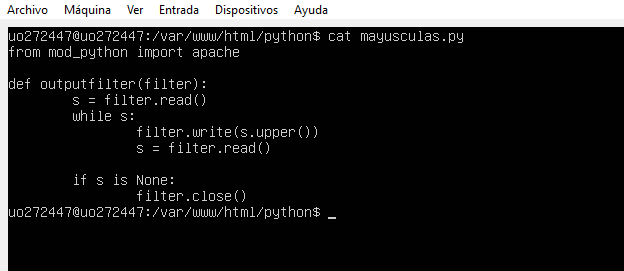


|  |  |
| --- | --- |
| **2.5** | *Crear un filtro de salida con Python que transforme las páginas de alguna forma* |

modificamos la configuración del directorio de la siguiente forma

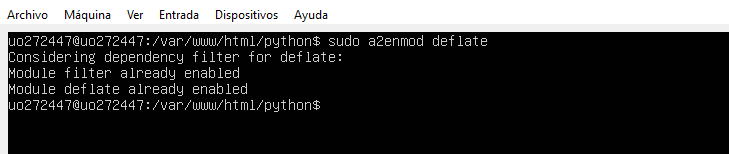


Se crea el programa en Python con el filtro

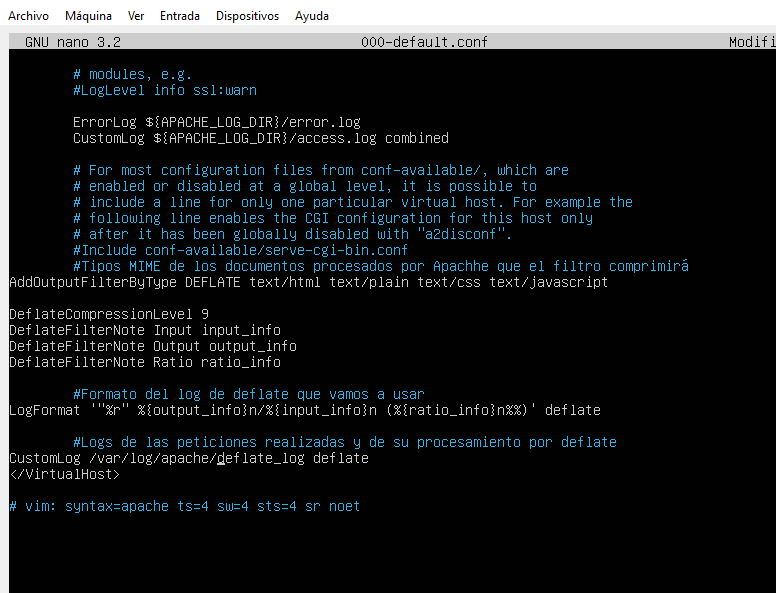


|  |  |
| --- | --- |
| **2.6** | *Habilitar mod\_deflate* |

Activar mod\_deflate, que ya esta activo

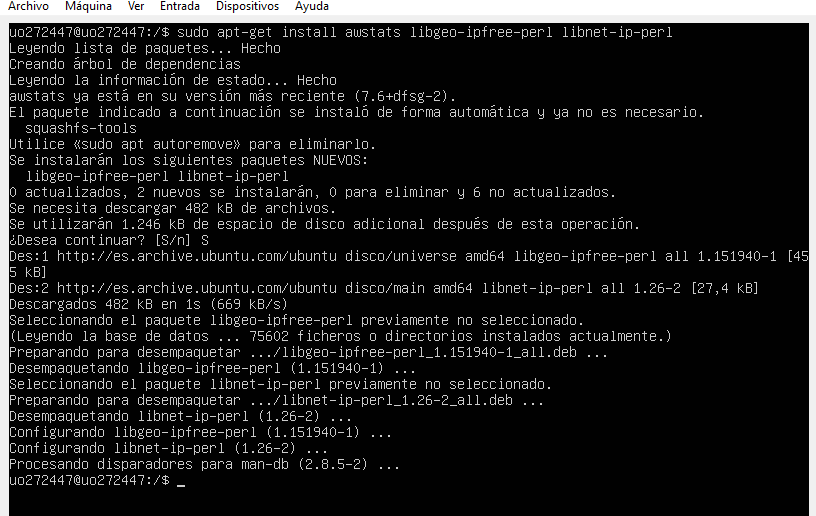


Modificar el fichero 000-default.conf

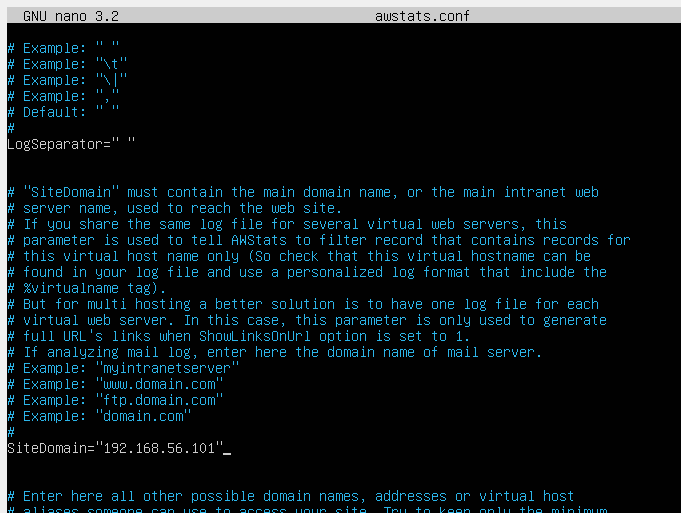


|  |  |
| --- | --- |
| **2.10** | *Ejecutar en análisis de ficheros de log con la herramienta awstats* |

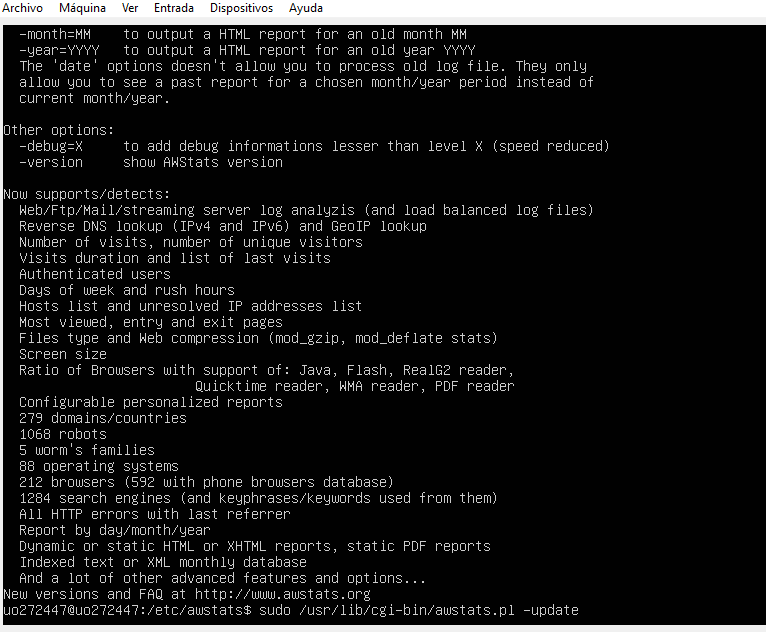
Instalar awstats



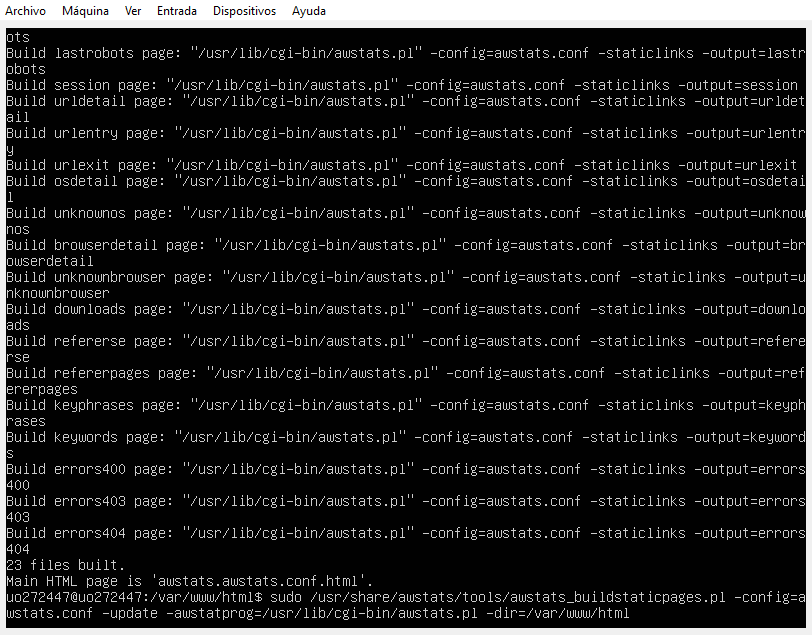
Se modifica el fichero awstats.conf para indicarle nuestra dirección



Actualizacion de los datos de awstats, ya que es la primera vez que lo arrancamos



Se ejecuta el comando para generar la información



Vemos a través del html generado el análisis exhaustivo que ha realizado la herramienta

