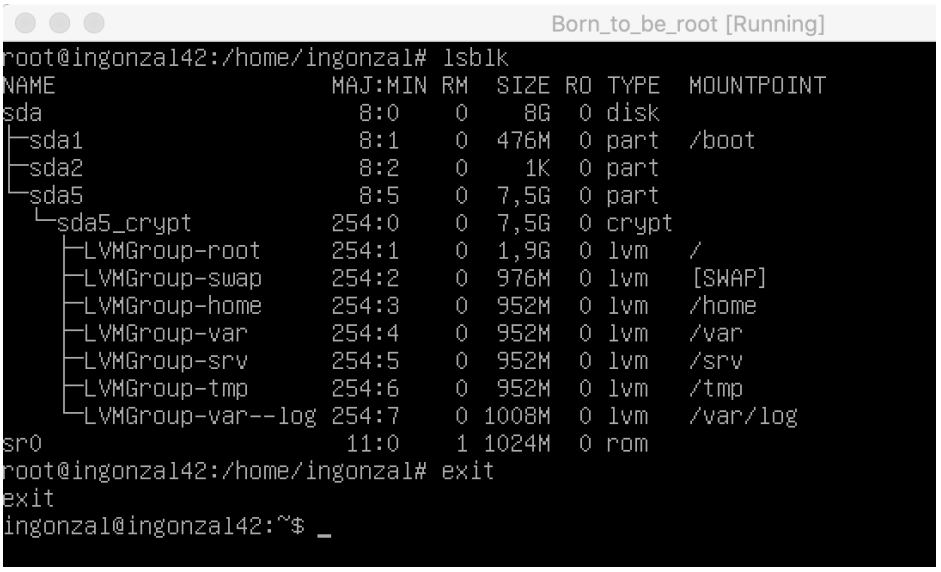
Born to be root

LSBLK



Diferencias Debian y CentOS:

Referencia 1.
Referencia 2.

- La discusión entre una distribución Red Hat (generalmente Centos) o una distribución Debian (generalmente Ubuntu Server o el propio Debian), es una discusión habitual cuando se adquiere un servidor dedicado o virtual en Linux.
- Utilizar Centos o Debian es una elección muy personal que, básicamente, depende de la destreza que el usuario tenga del entorno linux. Usabilidad y compatibilidad con servidores web.

CentOS - Community Enterprise Operating System.
Centos es que su sistema operativo es muy estable, evitando así los cuelgues que dejan el sistema completamente bloqueado.
La principal razón es que Centos es una derivación de otro sistema operativo comercial Red Hat Enterprise Server, por supuesto también basado en linux. El sistema operativo gratuito ha respetado básicamente todo el diseño del comercial, por lo que su calidad es muy elevada porque solo se encarga de ejecutar las versiones más estables, minimizando al máximo el riesgo de bloqueos del sistema, con lo que ello supone para el trabajo cotidiano.
Si hay una cosa cierta y una ventaja contundente que va ligada a Centos es la velocidad a la que trabaja este sistema operativo, mucho más superior que cualquiera de los otros sistemas basados en linux, sin excepciones ni competidores reales.
Asimismo, los programas de este sistema operativo son menos susceptibles de tener gusanos que bloqueen o que ataquen a la seguridad. De tenerlos, la velocidad se vería muy afectada por ralentizaciones y cuelgues, algo que no es muy poco probable, por no decir que es prácticamente imposible, con Centos.
Al margen de todas las buenas características que presenta, se ha de concluir que su funcionalidad es bastante más reducida que la de otros sistemas de linux. Este hecho hace que sea una solución útil solo para usuarios que no precisen de un procesador potente, porque en ese caso no tiraría con

las capacidades de Centos. También se han de valorar sus limitaciones, bastante acusadas, en las calidades de audio y vídeo.

Debian - Se basa en su sentido comunitario, respaldado por el trabajo constante y la participación de desarrolladores y usuarios que componen una gran plataforma muy activa y dinámica. Son precisamente ellos quienes en realidad mantienen este sistema operativo basado en linux.
La instalación es tremendamente sencilla y Debian, de hecho, puede ser instalado desde un CD, desde un soporte externo portátil o desde la propia red, sin mayores complicaciones y a pesar de su fama en contra.
Viene acompañado de casi 30.000 elementos de software y se puede contar con la asistencia que incluye, de forma que se puede instalar el software prácticamente sin ninguna intervención por parte del usuario.
Sistema operativo de paquetes muy bien integrados, con una calidad muy superior a la de otros sistemas basados en linux. Todos los paquetes del software se encuentran en el mismo sitio y se han eliminado dependencias complejas que solo servirían para entorpecer la libertad con que se integra Debian, convertido por ello en un sistema más robusto y eficiente.
Debian facilita enormemente la actualización a nuevas versiones de software, una faceta muy aplaudida de este sistema operativo.
Debian posee también un buen sistema de seguimiento de errores que, además, es público.
Debian esgrime la estabilidad como una de sus grandes banderas y no es de extrañar.
Los responsables de este sistema operativo señalan su rapidez y ligereza como una de las grandes ventajas de Debian y confirman que se emplean emuladores que, a veces, llegan incluso a ser más rápidos que los propios originales.
Es necesario analizar tres secciones bien diferenciadas que reportan cierto nivel de complicación o dificultad, lo que puede inclinar la balanza hacia otro tipo de sistema operativo llegado el caso.
Debian acaba adoleciendo de prestaciones en relación con la disponibilidad de algunos paquetes de software más populares.
Cuando se llega al apartado de configuración de hardware adicional, las cosas se complican y, tanto es así, que se trabaja en la propia plataforma para poder paliar esta dolencia y que puedan llegar a ser más fáciles las configuraciones más habituales.
Por último, existen serias dificultades cuando se trata de instalar hardware demasiado novedoso, curioso o ya obsoleto.
Conclusión - Si se es un usuario poco avanzado en conocimiento de linux o si se prefiere un entorno más simple y sencillo, sin grandes complicaciones ni quebraderos de cabeza, Centos es mucho más básico y se instala sin muchos aspavientos y sin necesidad de perder demasiado tiempo en el apartado de la configuración.
Por lo demás, se llega a la conclusión de que la elección debe basarse sobre todo en sensaciones más subjetivas, que dependen de cómo se sienta el usuario operando con cada uno de los sistemas. Se puede decir que se trata de una cuestión de gustos.
Sí es cierto que Debian parece un proyecto mucho más vivo, que aún no ha tocado techo y cuya comunidad de usuarios, constituidos en plataforma, están en constante ebullición para encontrar soluciones a las carencias que en la actualidad se presentan en este sistema operativo basado en linux.

Diferencias aptitude y apt:

Referencia.

APT - No garantiza la compatibilidad hacia atrás con apt-get, pero prácticamente todos los comandos tienen su equivalente, aunque corrige ciertos errores de diseño de la aplicación original, como por ejemplo que estructura mejor la información de los paquetes y agrega información extra al apt-cache que se usaba en el sistema clásico.
APTITUDE - Por su parte, aptitude gestiona el marcado de paquetes también, y dispone de una interfaz opcional de curses que a muchos les facilita el trabajo en la terminal.

Seguridad :

Referencia 1.
Referencia 2.

SELinux - Security-Enhanced Linux (SELinux) es un módulo de seguridad para el kernel Linux, que proporciona el mecanismo para implementar una políticas de control de acceso de tipo Control de acceso obligatorio (MAC) y control de acceso basado en roles. Se trata de un conjunto de modificaciones del núcleo y herramientas de usuario que pueden ser agregadas a diversas distribuciones Linux. Su arquitectura se enfoca en separar las decisiones de las aplicaciones de seguridad de las políticas de seguridad mismas y racionalizar la cantidad de software encargado de las aplicaciones de seguridad.45 Los conceptos clave que soportan SELinux pueden ser trazados a diversos proyectos previos de la Agencia de Seguridad Nacional de Estados Unidos.
** Una aplicación supervisada por SELinux posee acceso únicamente a los recursos que necesita, los cuales están descritos en una política de seguridad para dicho proceso. El acceso a procesos, puertos, archivos y directorios está controlado mediante reglas definidas en dicha política, es decir, el módulo de seguridad del kernel (SELinux), autoriza o deniega todas las operaciones del sistema con base en unas políticas de seguridad que definen por completo qué recursos del sistema pueden acceder las aplicaciones individuales y con qué privilegios.

AppArmor - ("Application Armor") es un módulo de seguridad del kernel Linux que permite al administrador del sistema restringir las capacidades de un programa. Para definir las restricciones asocia a cada programa un perfil de seguridad. Este perfil puede ser creado manual o automáticamente. Complementa el modelo tradicional de control de acceso discrecional de Unix (DAC) proporcionando el control de acceso obligatorio (MAC). Está implementado utilizando el framework del núcleo Linux Security Modules.

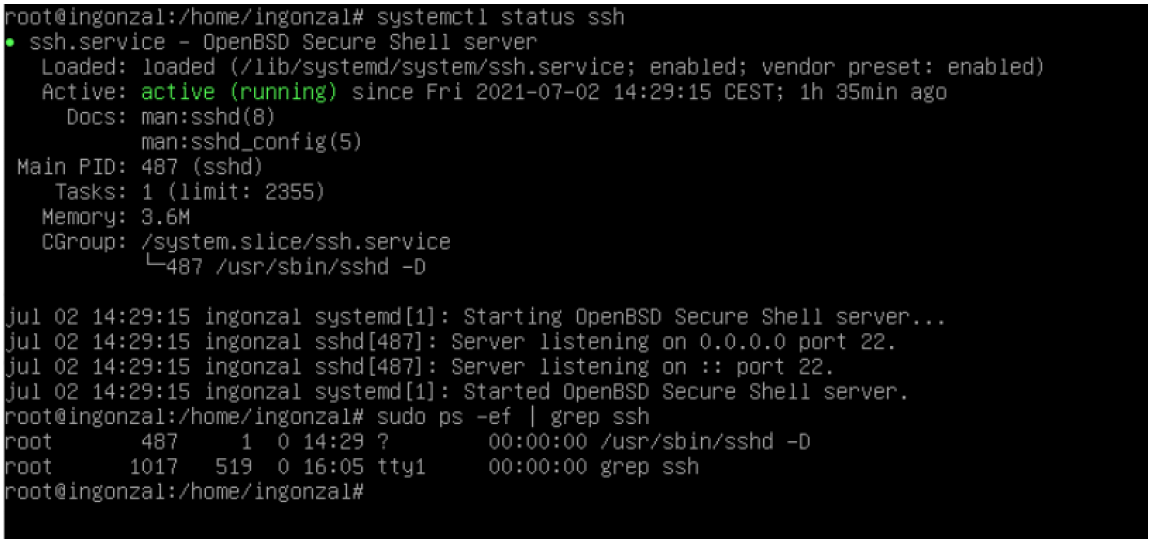
Diferencias SELinux y AppArmor:

SELinux está orientado a proteger el sistema completo, todos los objetos del sistema (ficheros, objetos IPC, redes, ...). Sin embargo AppArmor solo está orientado al sistema de ficheros.
AppArmor implementa una política centrada en la tarea, lo que significa que los atributos de control de acceso están vinculados a las tareas. Las reglas con las restricciones para cada aplicación están definidas en las llamadas perfiles de tareas. Estos pueden incluir habilidades para manipular ficheros específicos, usar red o montar dispositivos. El control de acceso está basado en rutas (de directorios, archivos, puertos,...) que se especifican. Programas que no tienen un perfil definido ejecutan sin restricciones por su parte (sólo las restricciones del sistema de permisos UNIX).
SELinux implementa una política centrada en los objetos del sistema. A cada objeto del sistema (proceso, puerto, archivo, directorio,..) se le asigna una etiqueta (contexto de seguridad), que será utilizada por la política de seguridad para determinar el acceso a este. El contexto de seguridad incluye distintos atributos como la identidad del usuario, rol, tipo, nivel,...
Ambos sistemas usan el principio de "denegación por defecto" (todas las operaciones son denegadas a menos que esté específicamente permitida por la política). Sin embargo, cada uno lo aplica de distinta manera.
AppArmor, al implementar una política centrada en la tarea, solo aplica el principio para aquellas tareas que controla.
SELinux aplica el principio en todo caso.
En general podemos decir que SELinux facilita un control de acceso de grado más fino que AppArmor. Por otro lado, AppArmor es más fácil de usar que SELinux ya que la configuración es más fácil de realizar (menos extensa).

SSH

Referencia.

Es un conjunto de estándares y protocolos de red que permite establecer una comunicación a través de un canal seguro entre un cliente local y un servidor remoto. Utiliza una llave pública para autenticar el servidor remoto y, de manera opcional, permitir al servidor remoto autenticar al usuario. SSH provee confidencialidad e integridad en la transferencia de los datos utilizando criptografía y MAC (Message Authentication Codes o Códigos de Autenticación de Mensaje).



Archivos de configuración:

/etc/ssh/sshd_config: Archivo principal de configuración del servidor SSH.
/etc/ssh/ssh_config: Archivo principal de configuración de los clientes SSH.
~/.ssh/config: Archivo personal de cada usuario. Contiene la configuración utilizada por los clientes SSH. Permite al usuario local utilizar una configuración distinta a la definida en el archivo /etc/ssh/ssh_config.
~/.ssh/known_hosts: Archivo personal de cada usuario. Contiene las firmas digitales de los servidores SSH a los que se conectan los clientes. Cuando éstas firmas cambian, se pueden actualizar ejecutando el comando ssh-keygen -R, pasando el nombre del anfitrión o la IP del anfitrión como argumento. Este comando elimina la entrada correspondiente en el archivo ~/.ssh/known_hosts y, permite añadir de nuevo al anfitrión con una nueva firma digital.
La sintaxis genérica sería:
$ ssh-keygen -R nombre_o_ip_del_servidor_SSH
~/.ssh/authorized_keys: Archivo personal para cada usuario. Contiene los certificados de los clientes SSH, para permitir autenticación hacia servidores SSH sin requerir contraseña.
Tras la instalación procederemos a la configuración del servidor, para ello, editaremos el fichero de configuración /etc/ssh/sshd_config.
NOTA: es importante hacer copia de seguridad del archivo /etc/ssh/sshd_config antes de realizar cualquier cambio.
$ sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config.bak
En nuestro ejemplo, editaremos el archivo /etc/ssh/sshd_config con el editor de texto plano nano, presente en casi todas las distribuciones Linux:
$ sudo nano /etc/ssh/sshd_config
Sobre el fichero de configuración por defecto, modificaremos las directrices establecidas en él y añadiremos aquellas directrices que no estén.

Directriz Port
De forma predeterminada, el servicio SSH escucha por el puerto 22. La directriz por defecto sería:
Port 22
Directriz ListenAddress
De forma predeterminada, el servicio SSH escuchará peticiones a través de todas las direcciones IP correspondientes a todas las interfaces de red del sistema. La directriz por defecto sería:
ListenAddress 0.0.0.0
Directriz Protocol
Existen dos versiones de SSH en cuanto a su protocolo de comunicación, SSHv1 y SSHv2. SSHv1 está en desuso pero todavía se incluye por compatibilidad. SSHv1 tiene varias vulnerabilidades conocidas, una de ellas en concreto, es un agujero de seguridad que potencialmente permite a un intruso insertar datos en la corriente de comunicación, por lo que su uso ya no es recomendable. Un error frecuente es dejar que el demonio SSH permita el uso de las dos versiones. Para evitar el uso de la versión SSHv1 y los posibles ataques a esta, basta con indicar en esta directriz que solo admita comunicaciones de SSH basadas en el protocolo SSHv2, el cual tiene un algoritmo de intercambio de claves mejorado y que no es vulnerable a los agujeros de seguridad de la versión SSHv1.
Protocol 2
Directriz LoginGraceTime
En esta directriz se establece el tiempo, en segundos, durante el cual la pantalla de login estará disponible para que el usuario introduzca su nombre de usuario y contraseña, si no lo hace durante ese período de tiempo el login se cerrará, evitando así dejar por tiempo indeterminado pantallas de login sin que nadie las use, o que alguien este intentando mediante un script adivinar un usuario y su contraseña. Si el valor es 0, no hay límite de tiempo para que un usuario se autentique, lo cual no es recomendable ya que de esta forma un atacante podría utilizar ataques de fuerza bruta o usando métodos de diccionario para adivinar la contraseña, por lo tanto no es recomendable dejar esta directriz a 0. En nuestro ejemplo lo estableceremos en 40 segundos.
Directriz PermitRootLogin
Probablemente sea la directriz de seguridad más importante que podemos establecer para asegurar nuestro servidor SSH. En los sistemas Unix y Linux se crea por defecto al usuario root, lo que implica que ya conocemos la existencia de al menos un usuario, ¡y que usuario!, con privilegios de administrador. Muchos ataques de fuerza bruta se concentran en atacar al usuario root con la esperanza de que tenga una contraseña débil.
Sabiendo una parte de la ecuación (root) solo será cuestión de tiempo para que alguien con paciencia y suerte vulnere el sistema. En esta directriz denegamos el acceso al usuario root y por lo tanto, cualquier intento de ataque directo al usuario root será inútil.
Al denegar el acceso al usuario root, cada vez que necesitemos realizar tareas administrativas, accederemos como un usuario normal y una vez dentro, utilizando alguno de los comandos su o sudo podremos realizar dichas tareas administrativas. Por lo tanto, denegando el acceso al usuario root, el atacante tendrá que acertar tanto el nombre de un usuario del sistema como su contraseña, algo que disminuye notablemente la probabilidad de una intrusión.
PermitRootLogin no
Directriz StrictModes
En esta directriz se establece que sshd revisara los modos y permisos de los archivos de los usuarios y el directorio $HOME de el usuario antes de aceptar la sesión. Esto es normalmente deseable porque a veces algunos usuarios dejan sus directorios, accidentalmente, con permiso de escritura para cualquiera. El valor predeterminado es yes, por lo tanto, lo dejaremos con su valor predeterminado.
Directriz MaxAuthTries
El valor de esta directriz establece el máximo número de intentos de autenticación permitidos por conexión, es decir, la cantidad de veces que podemos equivocarnos al ingresar el usuario y/o contraseña. Una vez que los intentos alcanzan la mitad de este valor, las conexiones fallidas siguientes serán registradas. Después del máximo número de intentos se cerrará la conexión. Es

posible volver a intentarlo, pero el límite de intentos por vez evita ataques basados en la persistencia de la conexión.

Directriz MaxStartups

El valor de esta directriz establece el máximo número de conexiones simultaneas de login que permitirá el servidor SSH por cada IP que intente conectarse. Hay ataques muy efectivos que dividen el ataque en una gran cantidad de conexiones de login. Es decir, el atacante divide en una gran cantidad de logins los intentos por ingresar, aumentando sus posibilidades de adivinar antes al usuario y su contraseña.

También podemos utilizar la siguiente sintaxis:

MaxStartups 10:30:60

Donde:

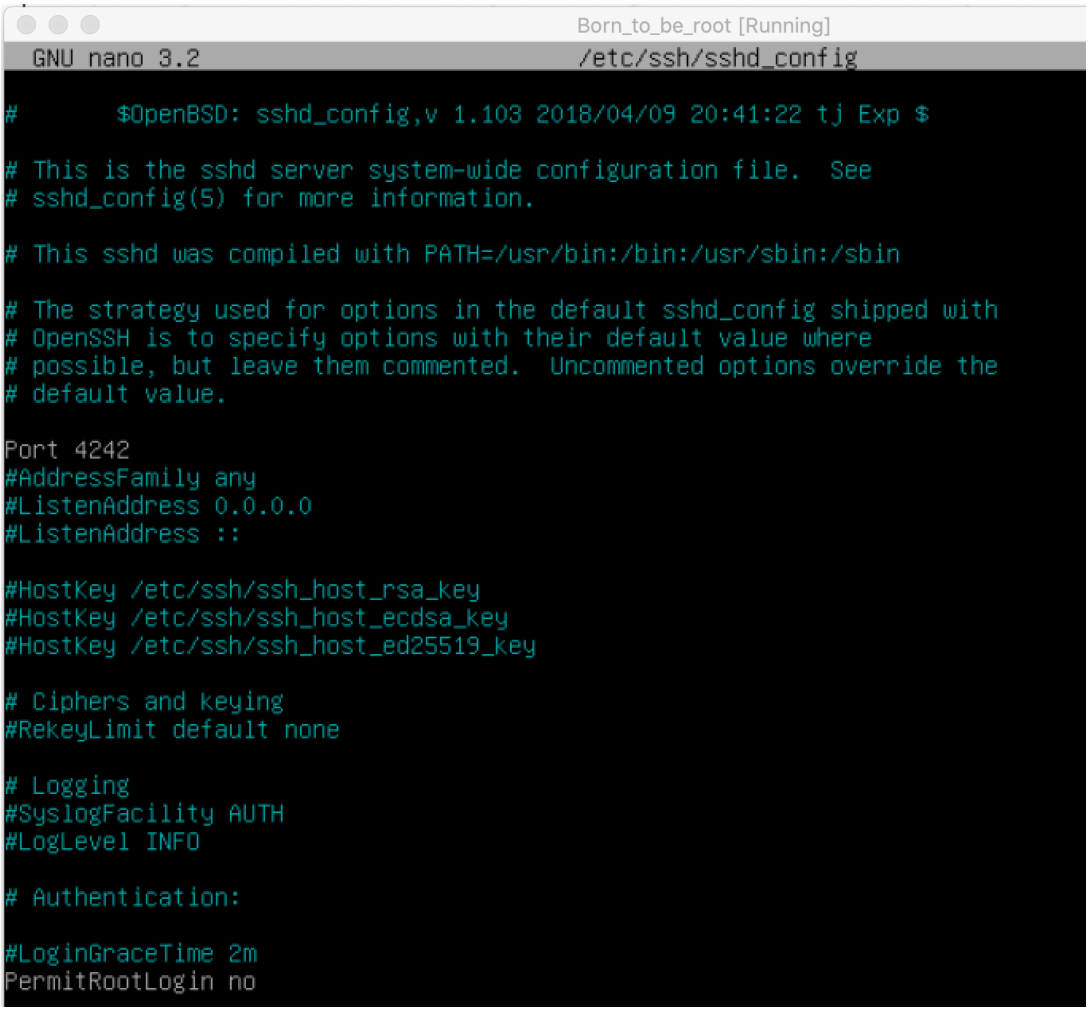
10: Número de conexiones no autenticadas antes de comenzar a caer.

30: Porcentaje de probabilidad de caer una vez que llegamos a 10.

60: Número máximo de conexiones en las que comenzamos a caer.

Directrices para autenticar con password

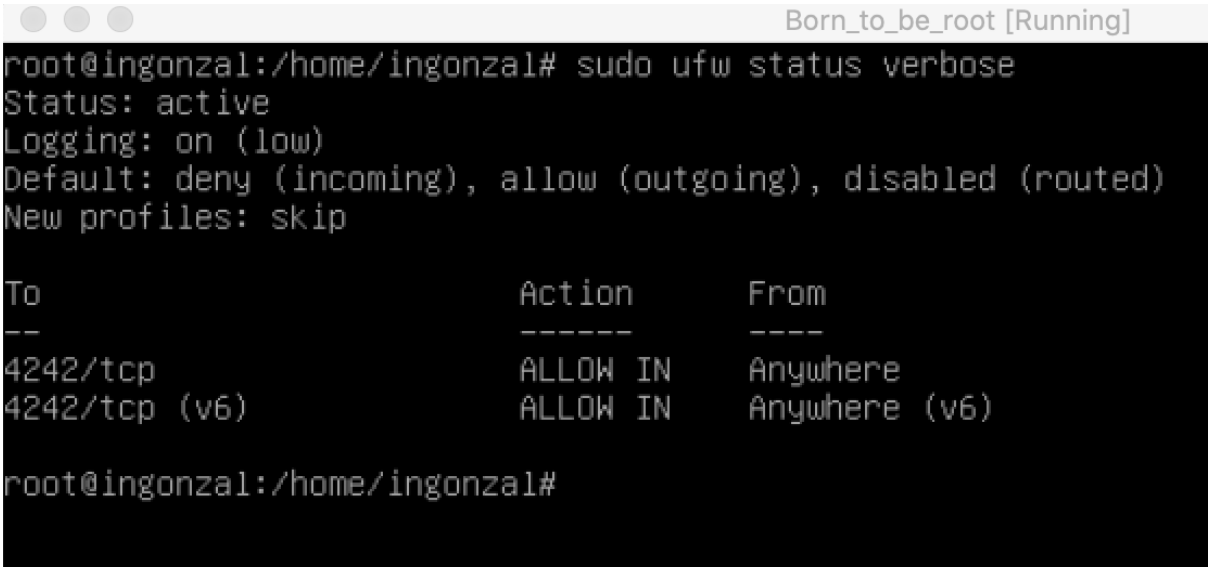
La directriz PasswordAuthentication habilita o deshabilita la autenticación con contraseñas. Por defecto está permitida la autenticación con contraseña. Si establecemos el valor no sólo se permitirá el acceso a través de firmas digitales. Es muy importante no cambiar el valor de esta directriz a no hasta haber instalado nuestra firma digital.



UFW

Referencia 1.
Referencia 2.

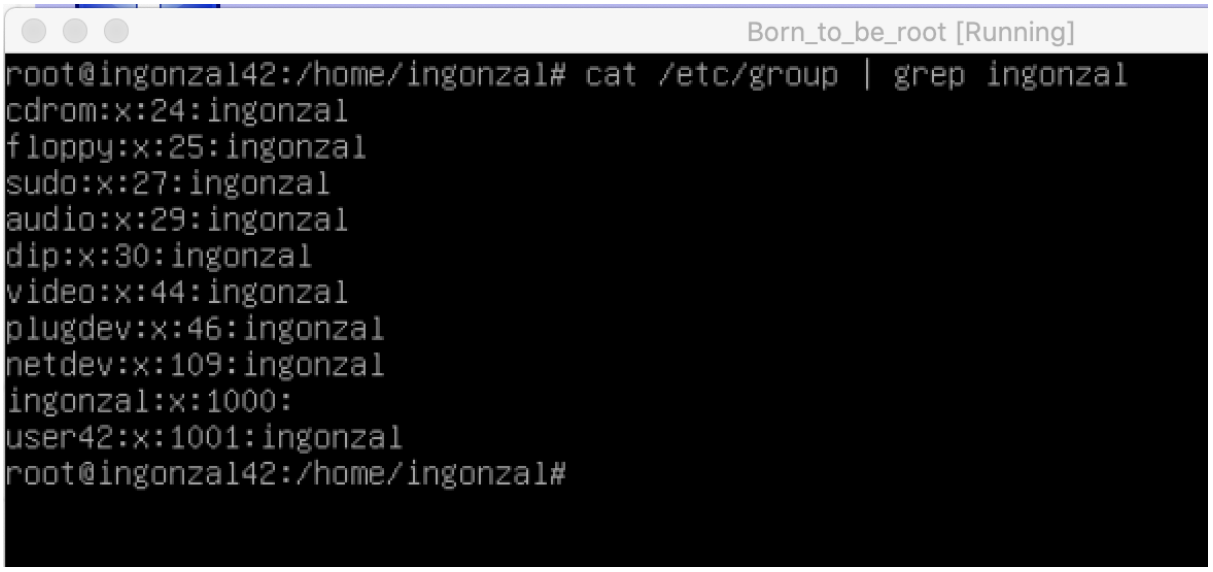
- sudo apt update
- sudo apt install ufw
- sudo ufw status verbose
- sudo ufw allow 4242/tcp
- sudo ufw enable
- sudo ufw status numbered



- El hostname de tu máquina virtual debe ser tu login terminado en 42 (por ejemplo, wil42). Deberás modificar este hostname durante tu evaluación. Referencia.

- Además del usuario root, un usuario con tu login como nombre debe existir.
- Este usuario debe pertenecer a los grupos user42 y sudo.
  Referencia 1.
  Referencia 2.

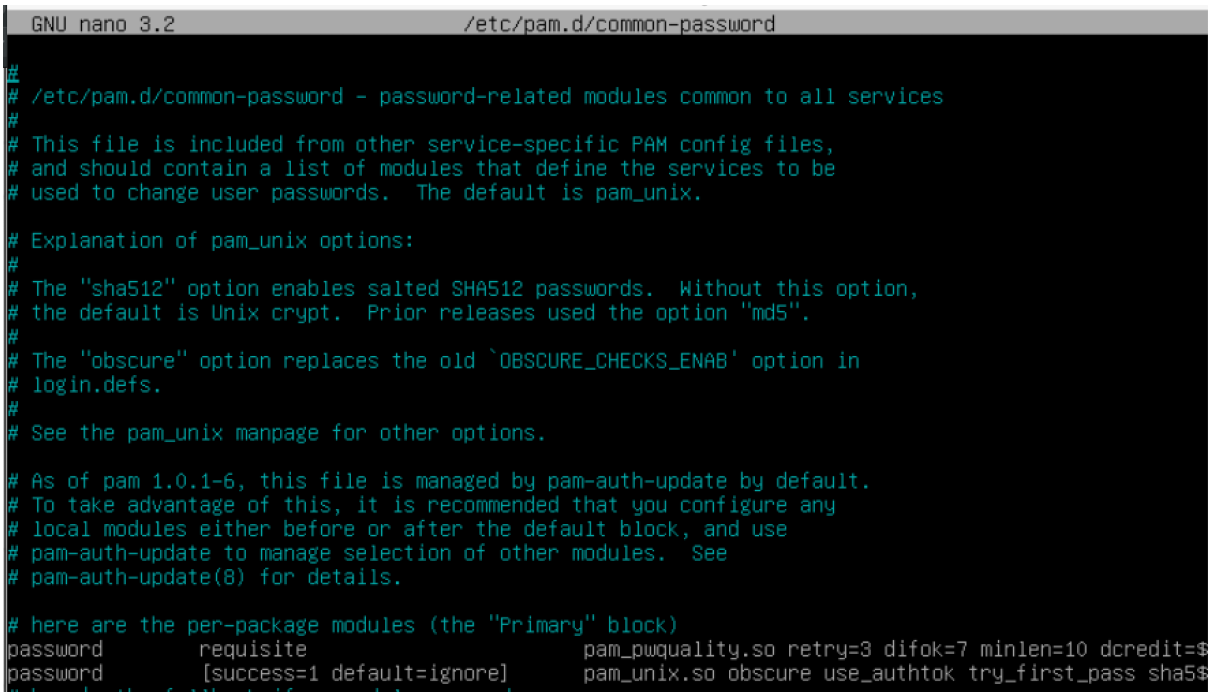
  - groupadd “nombredelgrupo”
  - gpasswd -a “nombredeusuario” “nombredegrupo”
  - cat /etc/group | grep “usuario”



- Debes implementar una política de contraseñas fuerte.
  Referencia 1.
  Referencia 2.
  Referencia 3.
  Referencia 4.
  - sudo apt install libpam-cracklib
  - sudo nano /etc/login.defs
    - Tu contraseña debe expirar cada 30 días.
    - El número mínimo de días permitido antes de modificar una contraseña deberá ser 2.
    - El usuario debe recibir un mensaje de aviso 7 días antes de que su contraseña expire.

- La contraseña no puede contener el nombre del usuario.
- La contraseña debe tener al menos 7 caracteres que no sean parte de la antigua contraseña. Por si te lo preguntas, la contraseña de root no tiene que seguir esta norma.
- Evidentemente, tu contraseña para root debe seguir esta política.
- Tu contraseña debe tener como mínimo 10 caracteres de longitud. Debe contener una mayúscula y un número. Por cierto, no puede tener más de 3 veces consecutivas el mismo carácter.
- La contraseña no puede contener el nombre del usuario.

- La contraseña debe tener al menos 7 caracteres que no sean parte de la antigua contraseña. Por si te lo preguntas, la contraseña de root no tiene que seguir esta norma.
- Evidentemente, tu contraseña para root debe seguir esta política.



- sudo nano /etc/sudoers

  Referencia.

  - - secure_path = ":snap/bin". Esto se hace por seguridad
  - - badpass_mesaagge = mensaje de error cuando la contraseña es correcta.
  - - logfile = registro de veces que se utiliza sudo. Crear la carpeta “Mkdir /var/log/sudo/” y un archivo llamado “sudo.log”

Monitoing.sh

Referencia 1.
Referencia 2.
Referencia 2.
• La arquitectura de tu sistema operativo y su versión de kernel.
echo "#Arquitecture: " $(uname -a)
• El número de núcleos físicos.
echo "#CPU physical:" $(lscpu | awk ' NR==5 {print $2}')
echo "#CPU physical:" $(lscpu | grep CPU\(s\) | awk ' {print $2}')
• El número de núcleos virtuales.
echo "#vCPU :" $(lscpu | awk ' NR==9 {print $2}')
echo "#vCPU :" $(lscpu | grep Socket\(s\) | awk '{print $2}')
• La memoria RAM disponible actualmente en tu servidor y su porcentaje de uso.
free --mega | awk 'NR==2{printf "Memory Usage: %s/%sMB (%.2f%%)\n", $3,$2,$3*100/$2 }'
• La memoria disponible actualmente en tu servidor y su utilización como un por- centaje.
df -h | awk '$NF=="/"{printf "Disk Usage: %d/%dGB (%s)\n", $3,$2,$5}'
• El porcentaje actual de uso de tus núcleos.
top -bn1 | grep load | awk '{printf "CPU Load: %.2f%%\n", $(NF-2)}'
• La fecha y hora del último reinicio.
echo "#Last Boot:" $(who -b | awk ' {print $3, $4}')
• Si LVM está activo o no.
echo "#LVM use:" $(lvm pvdisplay | grep Allocatable | awk '{print $2}')
• El número de conexiones activas.
echo "#Connetions TCP:" $(ss -s | grep TCP | awk 'NR==2 {printf "%d ESTABLISHED\n", $3}')
• El número de usuarios del servidor.
echo "#User log:" $(who | wc -l)
• La dirección IPv4 de tu servidor y su MAC (Media Access Control)
echo "#Network: IP" $(hostname -I) $(ip a | grep link/ether | awk '{printf " (%s)\n", $2}')
• El número de comandos ejecutados con sudo.
echo “#Sudo : $(cat /var/log/sudo/sudo.log | wc -l) cmd”

#Arquitecture
echo "#Architecture:" $(uname -a)
#CPU Physical
echo "#CPU physical:" $(lscpu | awk ' NR==5 {print $2}')
#Virtual CPU
echo "#vCPU :" $(lscpu | grep Socket\(s\) | awk '{print $2}')
#Memory Usage
free --mega | awk 'NR==2{printf "#Memory Usage: %s/%sMB (%.2f%%)\n", $3,$2,$3*100/$2 }'
#Disk Usage
df -h | awk '$NF=="/"{printf "#Disk Usage: %d/%dGB (%s)\n", $3,$2,$5}'
#CPU Load
top -bn1 | grep load | awk '{printf "#CPU Load: %.2f%s\n", $(NF-2), "%"}'
#Last Boot
echo "#Last boot:" $(who -b | awk ' {print $3,$4}')
#LVM
echo "#LVM use:" $(lvm pvdisplay | grep Allocatable | awk '{print $2}')
#Connections TCP
echo "#Connetions TCP:" $(ss -s | grep TCP | awk 'NR==2 {printf "%d ESTABLISHED\n", $3}')

```
#User log
echo "#User log:" $(who | wc -l)
#Network IP
echo "#Network: IP" $(hostname -I) $(ip a | grep link/ether | awk '{printf " (%s)\n", $2}')
#Sudo
echo “#Sudo :  $(cat /var/log/sudo/sudo.log | wc -l) cmd”
```

Referencia Contab

- Crontab:
    - */10 * * * * /root/monitoring.sh | wall

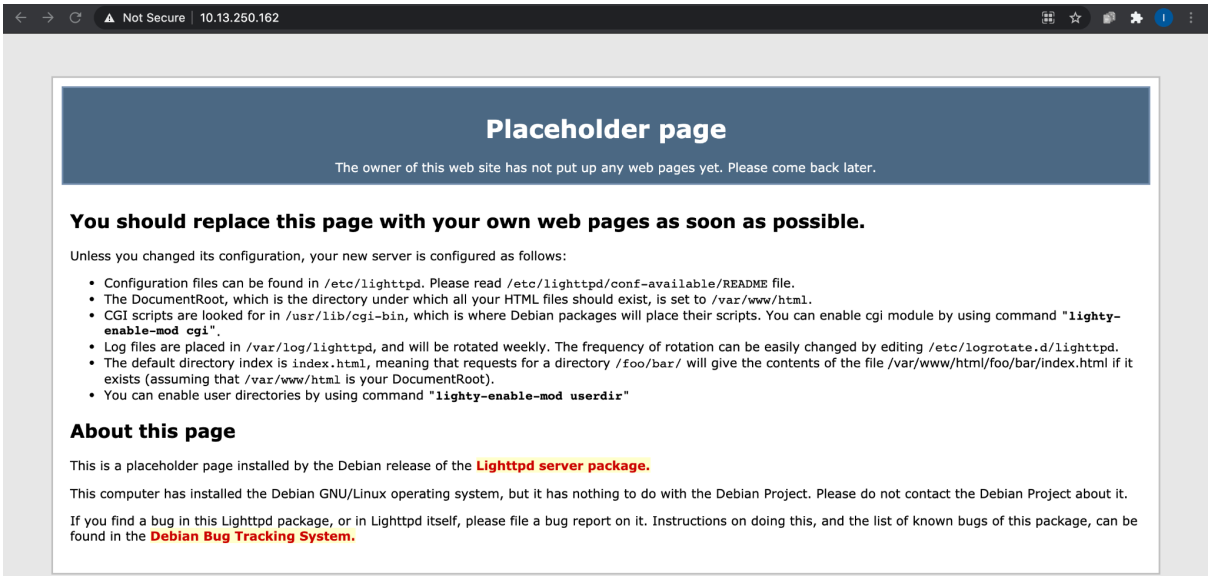
- Detener cron
    - "/etc/init.d/cron stop"

BONUS

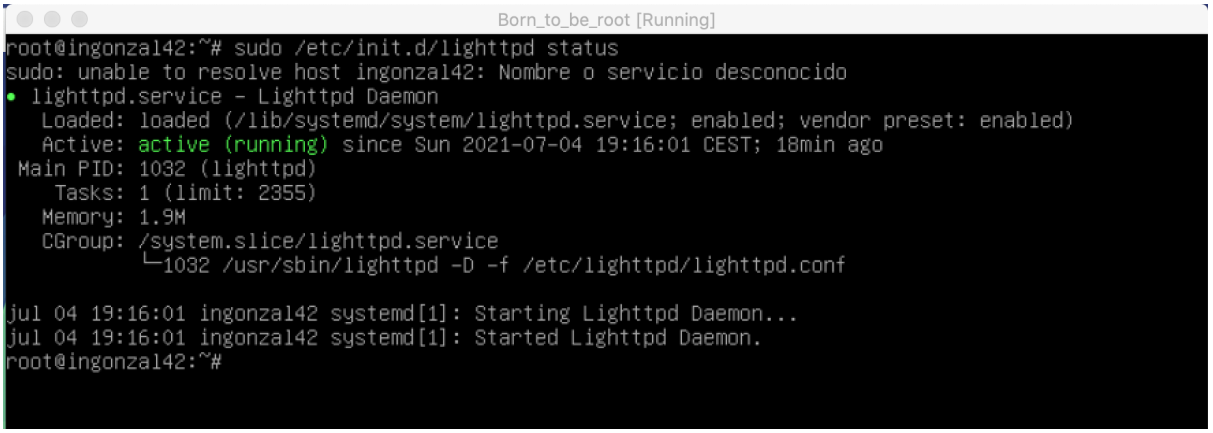
Referencia 1 .
Referencia 2 .
Referencia 3 .

- **Lighttpd** es un servidor web donde la característica principal de este, es su menor uso de cpu y memoria, lo que le hace perfecto para entornos donde, necesitamos una gran velocidad de proceso o donde las características del servidor no son muy buenas.

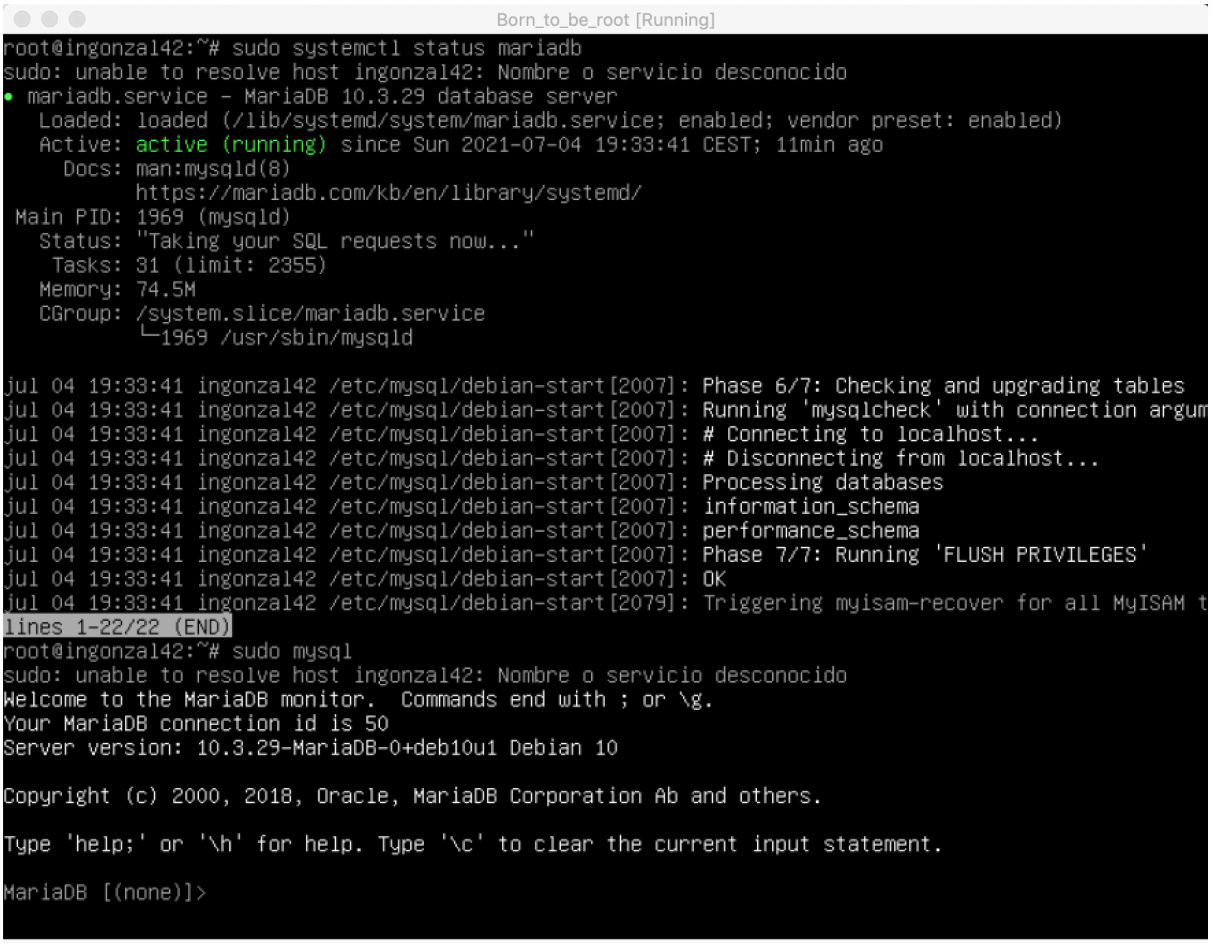
  - sudo apt install lighttpd
  - nmap -sT localhost lighttpd = puerto 80
  - sudo ufw allow 80
  - ip a ip inet = http://10.13.250.162/



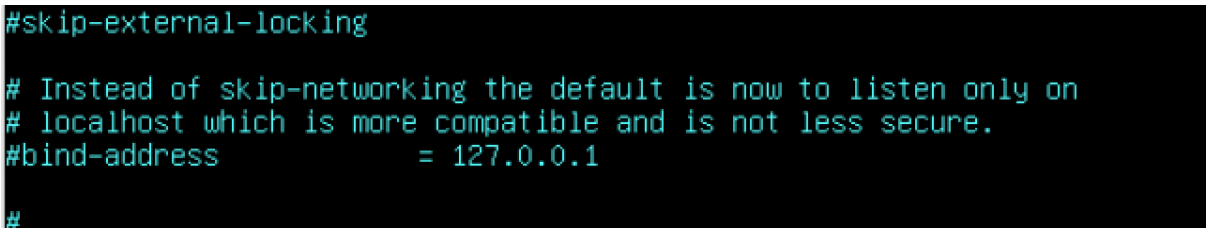
  - sudo /etc/init.d/lighttpd status
  - systemctl status lighttpd



- **Mariadb**
- sudo apt install -y mariadb-server
- sudo /etc/init.d/mariadb status
- systemctl status mariadb
- ~$ sudo mysql
- > grant all privileges on *.* to ingonzal identified by 'XXXXXXXX' with grant option;
- > flush privileges;

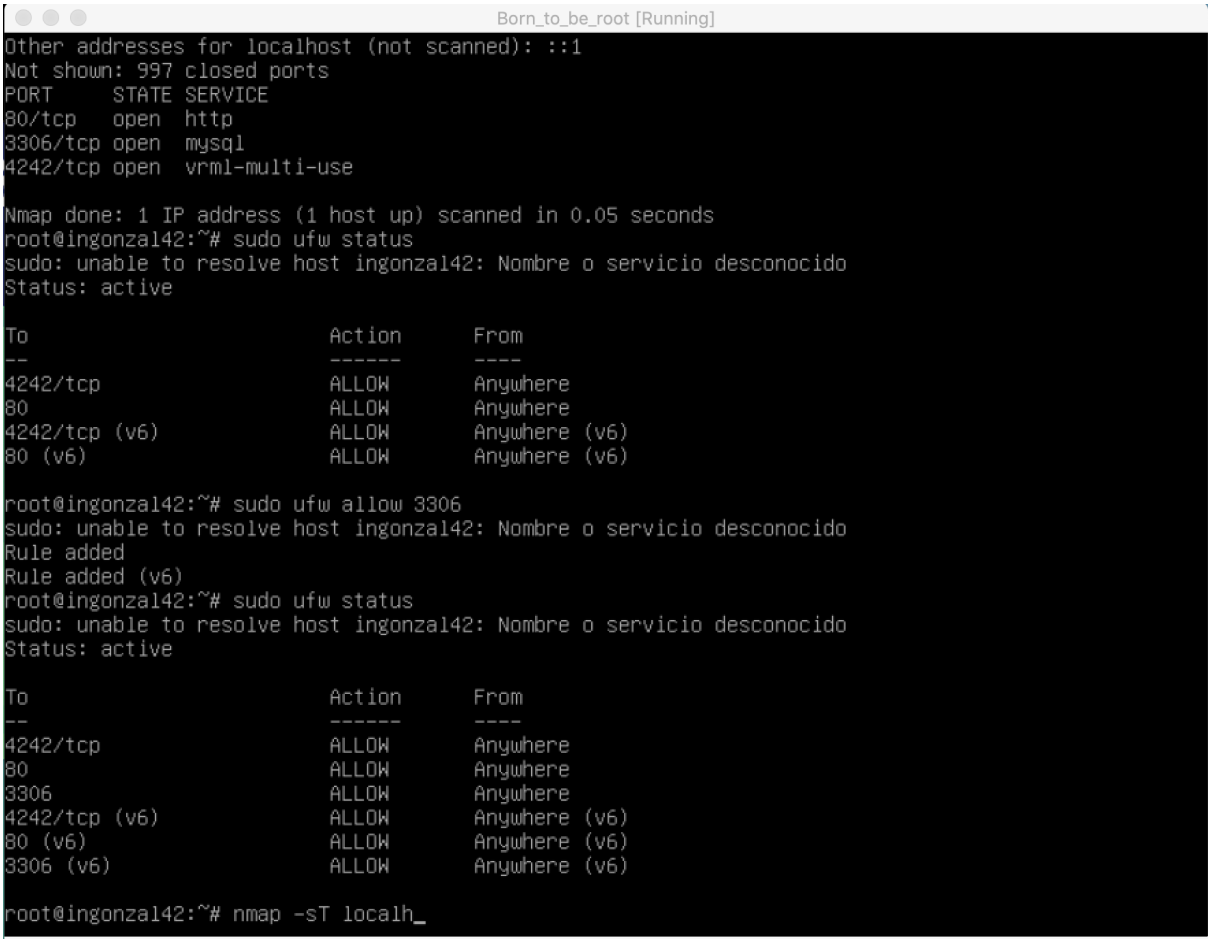


- sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf
- #bind-address = 127.0.0.1



- sudo systemctl restart mariadb

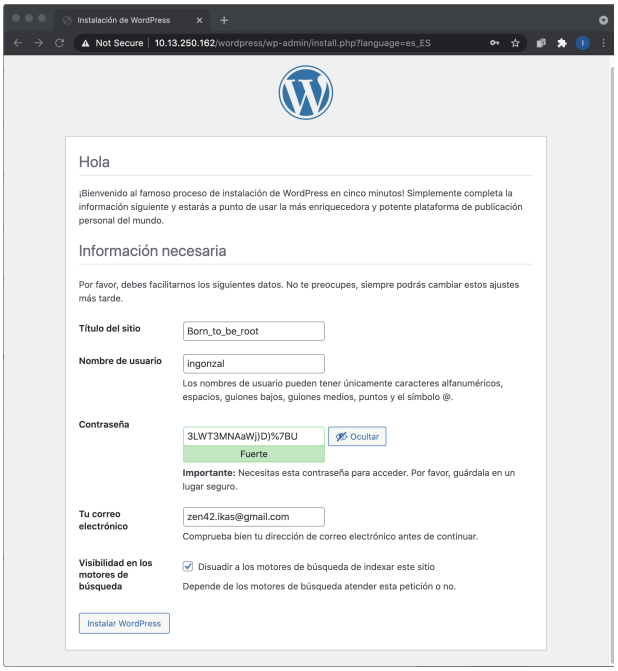
- nmap -sT localhost
- sudo ufw status
- sudo ufw allow 3306



- **PHP**
- sudo apt install php
- php -v

- **Wordpress**
  Referencia.
- wget https://es.wordpress.org/latest-es_ES.tar.gz
- sudo tar xf latest-es_ES.tar.gz -C /var/www/html
- sudo chown www-data:www-data /var/www/html/wordpress/ -R
- ~$ mysql -u root -p
- > create database wordpress charset utf8mb4 collate utf8mb4_unicode_ci;
- > create user wordpress@localhost identified by 'XXXXXXXX';
- > create user wordpress@localhost identified with mysql_native_password by 'ingonzal';
- > grant all privileges on wordpress.* to wordpress@localhost;
- > flush privileges;
- > exit
- sudo apt update
- sudo apt -y install php-bcmath php-curl php-imagick php-gd php-mbstring php-xml php-zip
- sudo apt -y install php7.4-bcmath php7.4-curl php7.4-imagick php7.4-gd php7.4-mbstring php7.4-xml php7.4-zip
- ~$ sudo systemctl reload apache2
- sudo a2enmod rewrite
- sudo nano /etc/apache2/conf-available/wordpress.conf
- El contenido será este bloque Directory:
- <Directory /var/www/html/wordpress>
- AllowOverride all
- </Directory>
- sudo a2enconf wordpress
- sudo systemctl restart apache2

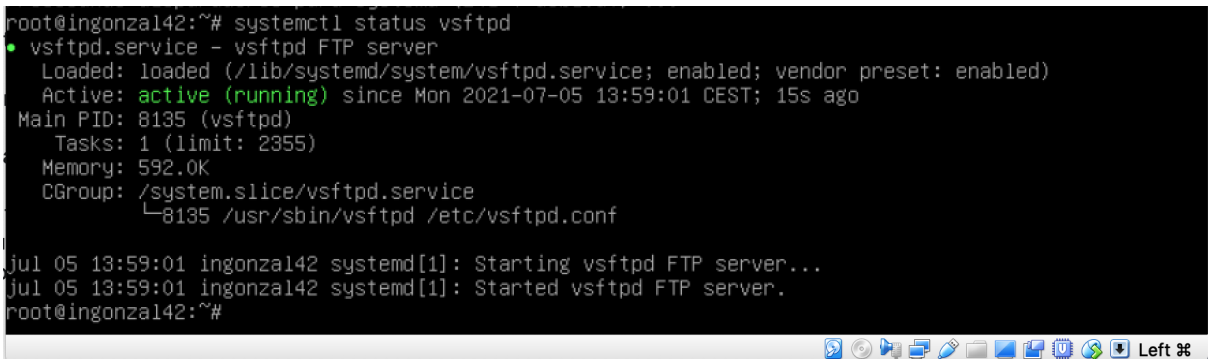
- pass: 3LWT3MNAaWj)D)%7BU
- Sitio Born_to_be_root



Servidor FTP

[Referencia.]

- sudo apt update
- sudo apt -y install vsftpd
- systemctl status vsftpd



- sudo ufw allow ftp
- sudo ufw allow ftp-data
- sudo ufw allow 20000:20100/tcp
- sudo apt install -y ftp
- sudo nano /etc/vsftpd.conf
    - ...
    - # Modo pasivo
    - pasv_enable=YES
    - pasv_min_port=20000
    - pasv_max_port=20100
    - ...
    - # Uncomment this to enable any form of FTP write command.
    - write_enable=YES

Mandatory

Project overwiew

- **How virtual machine works:**
  - Una máquina virtual permite establecer un entorno aislado para realizar pruebas controladas sobre un S.O o software específico.
- **Their choice of operating system:**
  - Debian. A pesar de ser una opción a priori algo más simple en su configuración ya que no posee una orientación tan clara a empresa. Es un SO tremendamente sólido que está englobado en un proyecto muy apoyado y sostenido por la comunidad.
- **The basic differences between CentOS and Debian**
  - Si se es un usuario poco avanzado en conocimiento de linux o si se prefiere un entorno más simple y sencillo, sin grandes complicaciones ni quebraderos de cabeza, Centos es mucho más básico y se instala sin muchos aspavientos y sin necesidad de perder demasiado tiempo en el apartado de la configuración.
  - Por lo demás, se llega a la conclusión de que la elección debe basarse sobre todo en sensaciones más subjetivas, que dependen de cómo se sienta el usuario operando con cada uno de los sistemas. Se puede decir que se trata de una cuestión de gustos.
  - Sí es cierto que Debian parece un proyecto mucho más vivo, que aún no ha tocado techo y cuya comunidad de usuarios, constituidos en plataforma, están en constante ebullición para encontrar soluciones a las carencias que en la actualidad se presentan en este sistema operativo basado en linux.
- **The purpose of virtual machine**
  - Establecer un entorno aislado para realizar pruebas controladas sobre un S.O o software específico.
- **Difference between aptitude and apt, and what APPArmor is.**
  - **APT -** No garantiza la compatibilidad hacia atrás con apt-get, pero prácticamente todos los comandos tienen su equivalente, aunque corrige ciertos errores de diseño de la aplicación original, como por ejemplo que estructura mejor la información de los paquetes y agrega información extra al apt-cache que se usaba en el sistema clásico.
  - **APTITUDE -** Por su parte, aptitude gestiona el marcado de paquetes también, y dispone de una interfaz opcional de curses que a muchos les facilita el trabajo en la terminal.
- **Display information.**
  - ok.

Simple setup

- **Ensure that the machine does not have a graphical environment at launch.**
  - ok
- **A password will be requested before attempting to connect to this machine.**
  - ingonzal
- **This user must not be root.**
  - ingonzal
- **Check that the UFW service has started.**
  - sudo ufw status verbose
  - sudo service ufw status
- **Check that the SSH service has started.**
  - sudo service ssh status
- **Check that the chosen operating system is Debian or CentOS.**
  - head -n 1 /etc/os-release

- uname -a

User

- **The subject requests that a user with the login of student evaluated is present on the virtual machine**
  - ingonzal
- **Check that it has been added and that it belongs to the “sudo” and “user42” groups.**
  - cat /etc/group | grep sudo
  - cat /etc/group | grep user42
- **Make sure the rules imposed in the subject concerning the password policy have been put in place by the following steps.**
  - sudo nano /etc/login.defs
  - sudo nano /etc/sudoers
  - sudo chage -l “ingonzal”
- **First, create a new user.**
  - sudo useradd “prueba”
- **Assign it a password of your choice, respecting the rules.**
  - sudo passwd “prueba”
  - C0ntrasena
- **Create a group named “evaluating” and assign it to this user.**
  - groupadd “evaluating”
  - gpasswd -a “nombredeusuario” “nombredegrupo”
- **Check that this user belongs to the “evaluating” group.**
  - cat /etc/group | grep evaluating
- **Ask the student to explain the advantages of this password policy, as well as the advantages and disadvantages of its implementation.**
  - Se pretende establecer una política de contraseñas fuerte para dificultar los accesos no autorizados y los intentos de penetración en el sistema. En este caso se exige una longitud máxima de 10 y el uso de una mayúscula y de un número, que no repita el mismo carácter 3 veces y que no contenga el nombre del usuario. Esto junto con sudola obligatoriedad de cambio de contraseña cada 30 días y que no se pueda repetir parte de la contraseña anterior hace que sea relativamente seguro y aunque puede establecerse una política mucho más restrictiva, el hacer muy inmanejable esto para el usuario se convierte en una falla de seguridad en sí misma, obligando al usuario a apuntar las claves y dificultando su desempeño.

Hostname and partitions

- **Check that the hostname of the machine is correctly formatted**
  - @ingonzal
  - hostnamectl status
  - cat /etc/hostname
  - cat /etc/hosts
- **Modify this hostname by replacing the login with yours, then restart the machine.**
  - hostnamectl set-hostname
- **You can now restore the machine to the original hostname.**
  - hostnamectl set-hostname “nuevo”
  - vim /etc/hostname
  - vim /etc/hosts
- **View the partitions.**
  - lsblk
- **Compare the output with the example.**
  - Bonus example

- **How LVM works and what it is all about.**
    - LVM se utiliza para gestionar el espacio de almacenamiento. El LVM establece una capa lógica entre el sistema de archivos y las particiones del almacenamiento de datos utilizado. Esto le permite crear un sistema de archivos que abarque varias particiones y/o discos. De este modo, por ejemplo, se puede combinar el espacio de almacenamiento de varias particiones o soportes de datos.

        Ventajas del LVM

    - El LVM ofrece las siguientes ventajas adicionales:

    - Los volúmenes lógicos pueden ser creados, ampliados o reducidos durante la operación. No es necesario formatear los soportes de datos. Sin embargo, el tamaño del volumen lógico debe ajustarse manualmente después.

    - Los datos se pueden reorganizar durante el funcionamiento.

    - Los datos pueden distribuirse en varios soportes de datos. Esto puede aumentar significativamente el rendimiento de los datos.

    - Los datos pueden ser fácilmente replicados.

    - Se pueden tomar snapshots durante el funcionamiento.

Sudo

- **Check that the "sudo" program is properly installed on the virtual machine.**
    - apt -qq list sudo
- **Show assigning your new user to the "sudo" group.**
    - gpasswd -a "nombredeusuario" "nombredegrupo"
- **Explain the value and operation of sudo using examples of their choice.**
    - permite regular y otorgar permisos de administrador a cualquier usuario con la ventaja de que al solicitar la clave en cada invocación, nos permite asegurar que no se va a utilizar ningún comando delicado de forma irreversible, cosa que si hacemos desde el usuario root, lo realiza de forma automática.
- **Show the implementation of rules imposed by the subject.**
    - sudo nano /etc/sudoers
- **Verify that the "/var/log/sudo/" folder exists and has at least one file**
    - logfile = registro de veces que se utiliza sudo. "/var/log/sudo/" y "sudo.log".
- **Run a sudo command and verify if that file has been updated**.
    - ok

UFW

- **Check that the "UFW" program is properly installed on the virtual machine.**
    - apt -qq list ufw
- **Check that it is working properly.**
    - sudo service ufw status
- **Explain basically what UFW is and the value of using it.**
    - Es el firewall con el que configuramos por que puertos de comunicación permitimos que haya flujo de datos. Con esto permitimos que que unos u otros servicios estén a la escucha y podemos así hacer un servicio más seguro.
- **List the active rules in UFW. A rule must exist for port 4242.**
    - sudo service ufw status

- **Add a new rule to open 8080. Check the list.**
    - sudo ufw allow 8080
- **Delete this new rule.**
    - sudo ufw status numbered
    - sudo ufw delete “number”

SSH

- **Check that the SSH service is properly installed on the virtual machine**
    - apt -qq list ssh
- **Check that it is working properly.**
    - sudo service ssh status
- **Explain basically what SSH is and the value of using it.**
    - SSH o Secure Shell, es un protocolo de administración remota que le permite a los usuarios controlar y modificar servidores remotos a través de Internet por medio de un mecanismo de autenticación.
- **Verify that the SSH service only uses port 4242.**
    - vim /etc/ssh/sshd_config
    - nmap -sT localhost
- **log in with the newly created user.**
    - ssh ingonzal@10.13.250.162 -p 4242
    - sudo reboot
- **Make sure does not work with the root user**
    - ssh rootl@10.13.250.162 -p 4242

Script Monitoring

- **How their script works by showing you the code.**
    - ok
- **What “cron” is.**
    - Cron es un administrador de tareas de Linux que permite ejecutar comandos en un momento determinado, por ejemplo, cada minuto, día, semana o mes. Si queremos trabajar con cron, podemos hacerlo a través del comando crontab.
- **How the student being evaluated set up their script so that it runs every 10 minutes.**
    - crontab -e
    - */10 * * * * /root/monitoring.sh | wall
    - ok
- **Runs every minute.**
    - */1 * * * * /root/monitoring.sh | wall
    - /etc/init.d/crond restart
- **Stop running cron .**
    - /etc/init.d/crond stop

BONUS

- **Setting up partitions**
    - lsblk
- **Setting up Wordpress**
    - Wordpress
- **The free choice service**
    - FTP
    - ftp://10.13.250.162

ENTREGA

- cd /goinfre
- shasum Born2BeRoot.vdi >> signature.txt