Prueba de selección de SISTEMAS

RAFAEL DELGADO PEÑA

NOSOLOSOFTWARE | Avda. del Gran Capitán, 46, 14006 Córdoba

1 CONTENIDO

1. Manejo de la consola Linux 2

1.1 ¿Qué hace el siguiente comando: for u in $(cat /etc/passwd| cut -d: -f1); do crontab -l -u $u; done? 2

1.2 ¿Cómo cambiarías el propietario a la ruta /var/www/html y a todos sus ficheros para el usuario www-data? 3

1.3 Da permisos de lectura, escritura y ejecución para el usuario y grupo propietario del directorio /var/www/html, el resto de usuarios no deben tener permisos. 4

1.4 ¿Cómo obtendríamos el resultado por consola de la primera columna de un fichero con el siguiente contenido? 5

1.5 ¿Cómo darías permisos de sudo al usuario ejemplo1? 5

1.6 Escribe una tarea de cron 7

1.6.1 Se ejecuta únicamente los lunes cada 20 minutos 7

1.6.2 Guarda un histórico con la fecha y hora del sistema de cada ejecución en el fichero /var/log/ejemplo-cron.log 7

1.7 Comando para crear un usuario 8

1.7.1 Llamado ejemplo2 8

1.7.2 Su home debe estar en /var/www/html/ejemplo2 8

1.8 Escribe el comando para cambiar la contraseña del usuario ejemplo2 9

1.9 Comando para conectarnos pos ssh 10

1.9.1 Al servidor prueba.ejemplo.com 10

1.9.2 Con el usuario ubuntu 10

1.9.3 Puerto 2022 10

1.10 Añade una variable de entorno llamada prueba cuyo valor sea nosolosoftware para la consola actual 11

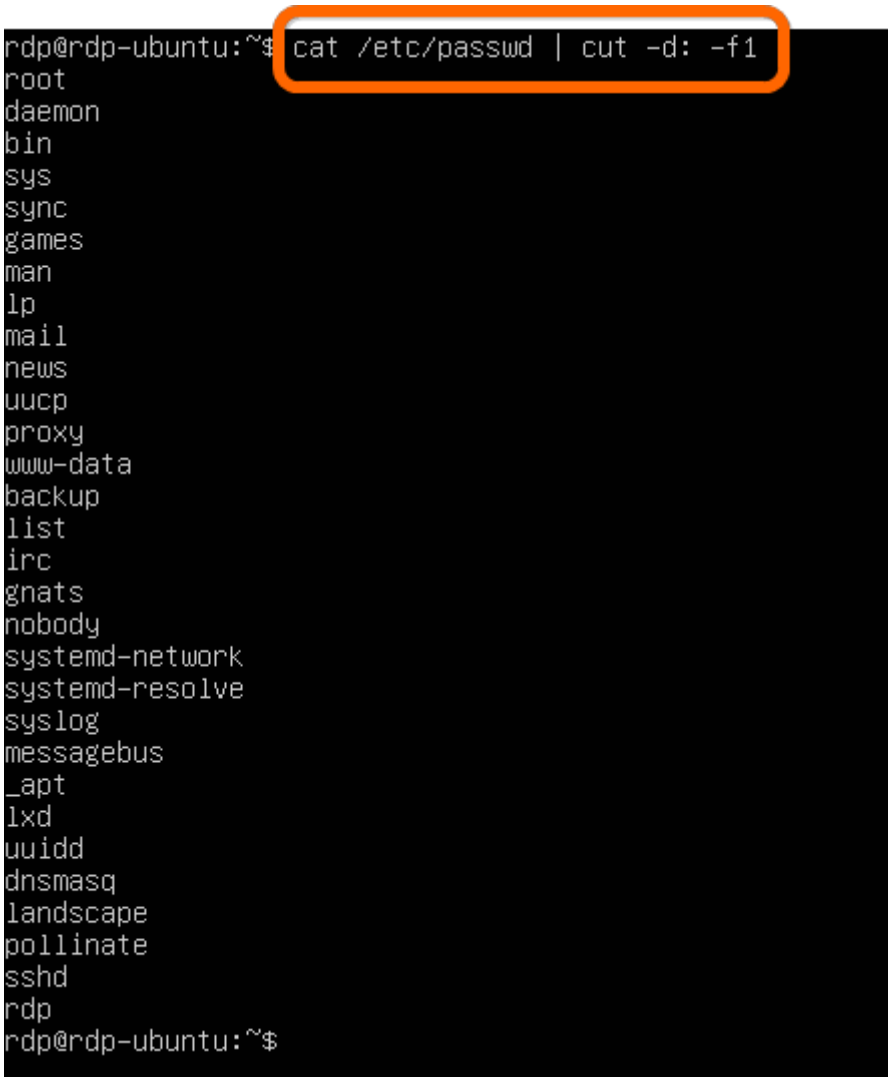
1.11 Comando para ver los registros mx de google.es 11

1.12 ¿Qué hace el comando find /home/ubuntu | grep –i id_rsa? 13

1. MANEJO DE LA CONSOLA LINUX

1.1 ¿QUÉ HACE EL SIGUIENTE COMANDO: FOR U IN $(CAT /ETC/PASSWD| CUT -D: -F1); DO CRONTAB -L -U $U; DONE?

Lo primero, vamos a analizar el comando que está entre paréntesis:



Muestra el nombre de cada usuario del contenido del fichero passwd. Con el comando cut –d: -f1 lo que hacemos es cortar mediante el delimitador ":" el primer campo, que es el nombre de cada usuario.

Voy a mostrar el contenido del fichero sin el comando cut para una mayor comprensión:

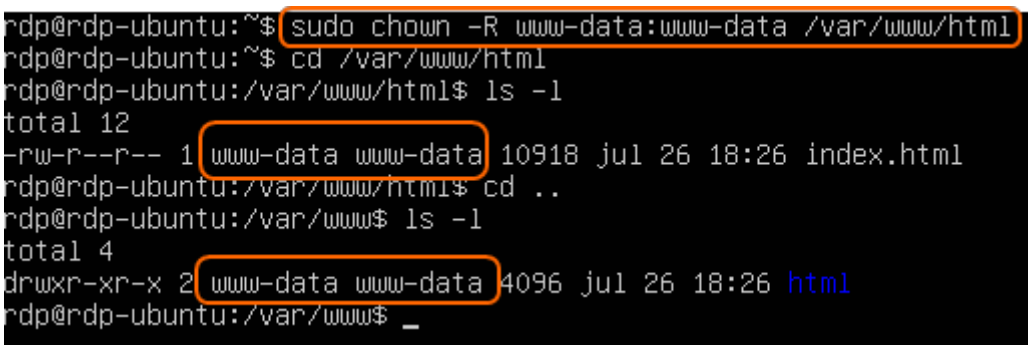
El argumento **-f1** lo que nos va hacer es cortarnos todo lo que tenga el primer dato separado por el delimitador **-d:** y así obtendremos el nombre de los usuarios.

Ahora nos vamos al principio del comando, que es **for u in**. Lo que empezamos a hacer aquí es que, por cada usuario (variable u, para asignar el nombre de usuario y guardarla en esta variable) que muestre el comando cat (parte del paréntesis), lístame las tareas que dicho usuario tiene. Veamos la parte final: **do crontab –l** (listar) **–u** (usuario) **$u** (contiene el nombre obtenido del comando cat).

En resumen, el comando muestra/lista las tareas que tiene asignadas cada usuario que se encuentre en el fichero /etc/passwd y se van mostrando una a una dependiendo de si los usuarios tienen tareas crontab asignadas o no.

1.2 ¿CÓMO CAMBIARÍAS EL PROPIETARIO A LA RUTA /VAR/WWW/HTML Y A TODOS SUS FICHEROS PARA EL USUARIO WWW-DATA?

Con el siguiente comando:

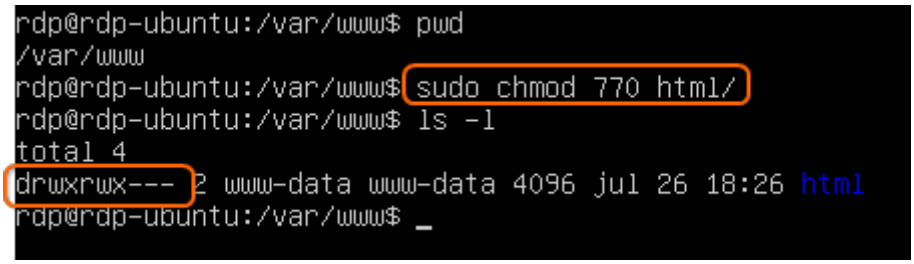


El argumento –R es para aplicar el cambio de propietario de forma recursiva, así, todos los ficheros que estén dentro del directorio html, cambiarán también de propietario.

Después del argumento –R, indicamos el **usuario:grupo** del cual queremos que pertenezca este directorio. Todo esto solo podremos hacerlo si poseemos permisos de súper usuario(sudo) o si somos el súper usuario (root).

1.3 DA PERMISOS DE LECTURA, ESCRITURA Y EJECUCIÓN PARA EL USUARIO Y GRUPO PROPIETARIO DEL DIRECTORIO /VAR/WWW/HTML, EL RESTO DE USUARIOS NO DEBEN TENER PERMISOS.

Lo haré con el siguiente comando:

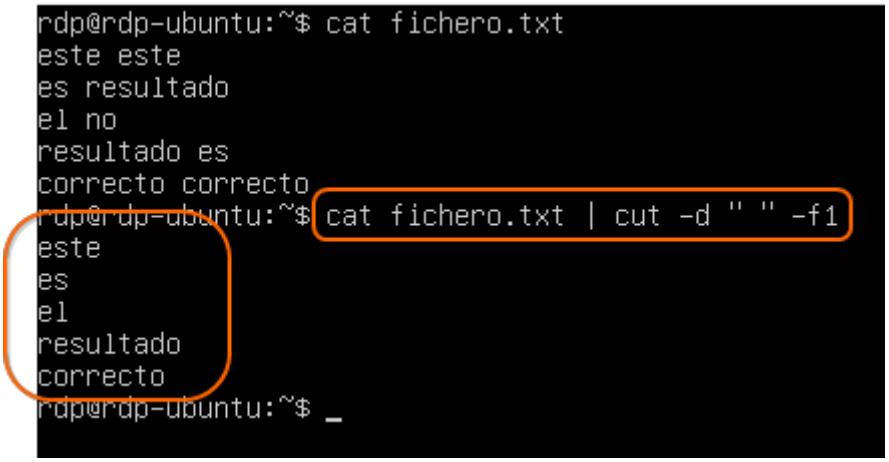


Con respecto al número 770, se divide en que cada número representa los permisos del usuario (7), grupo (7) y resto de usuarios (0). Los permisos son lectura, escritura y ejecución. Si nos vamos al formato binario, el 7 es: 111. El 1 indica que el permiso está activado, mientras que el 0 indica que está desactivado. Por lo tanto, el usuario tiene los permisos de lectura (1), escritura (1) y ejecución (1) activados (111) al igual que el grupo. El 0, como podemos observar indica que para el resto de usuarios los permisos están todos desactivados, ya que el 0 en binario se representaría en este caso, de esta forma: 000. Con lo cual estarían todos los permisos desactivados para el resto de usuarios, no podrán ni leer, ni escribir ni ejecutar los ficheros del directorio html.

1.4 ¿CÓMO OBTENDRÍAMOS EL RESULTADO POR CONSOLA DE LA PRIMERA COLUMNA DE UN FICHERO CON EL SIGUIENTE CONTENIDO?

```
este este
es resultado
el no
resultado es
correcto correcto
```

Sería con el siguiente comando:



Primero he creado y mostrado el fichero tal y como es.

Si observamos, está compuesto por dos columnas separadas por espacios en este caso.

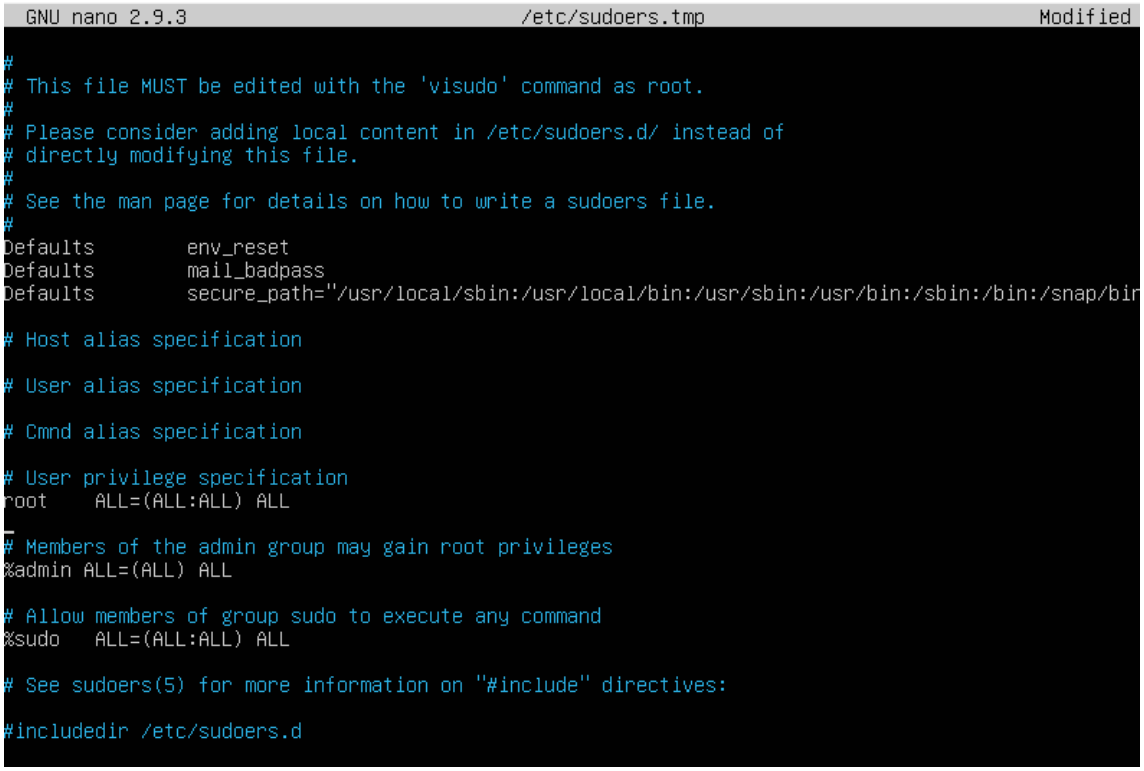
He usado el comando **cut** para filtrar el resultado por consola, de tal manera que me muestre la primera columna (-f1) e indicándole el delimitador (-d " ") que en este caso es un espacio.

1.5 ¿CÓMO DARÍAS PERMISOS DE SUDO AL USUARIO EJEMPLO1?

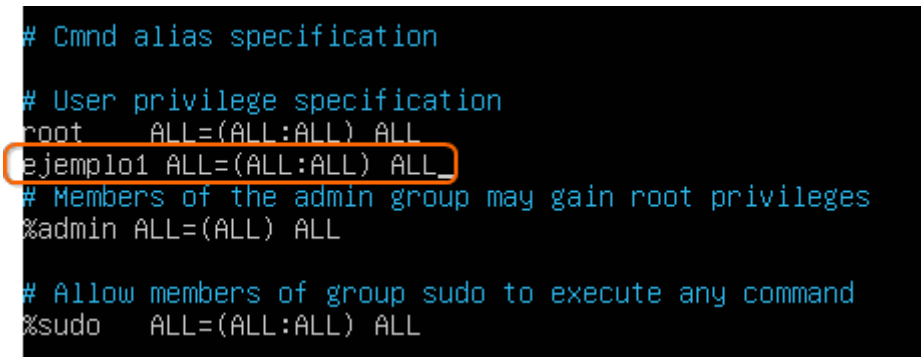
Se puede hacer de varias maneras, añadiendo a ejemplo1 al grupo de administradores o bien añadiendo a ejemplo1 en el fichero sudoers:

Escribimos en consola: sudo visudo.

Con este comando accedemos al archivo etc/sudoers:



Ahora escribimos la siguiente línea después del usuario root:



Vamos a analizar la línea:

- ejemplo1: es el nombre de nuestro usuario en este caso.
- Primer ALL: indica que va a aplicar a cualquier anfitrión.
- Segundo ALL: Indica que va a poder usar comandos de cualquier usuario.
- Tercer ALL: indica que va a poder usar comandos de cualquier grupo.
- Cuarto ALL: las reglas se aplican a todos los comandos.

Guardamos el fichero. Para que afecten los cambios es necesario cerrar sesión.

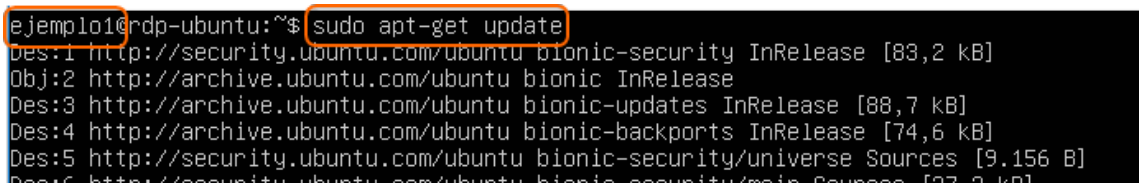
El usuario ejemplo1 ya dispone del comando sudo.

La otra forma es añadir al usuario ejemplo1 al grupo sudo:

El atributo –a es para añadir al grupo (-G) sudo al usuario ejemplo1.

Esto añadirá a ejemplo1 al grupo sudo.

El usuario ejemplo1 ya puede usar también el comando sudo:



Debemos mirar con lupa a qué usuarios añadimos a sudoers… Esto puede ser un problema de seguridad bastante grande.

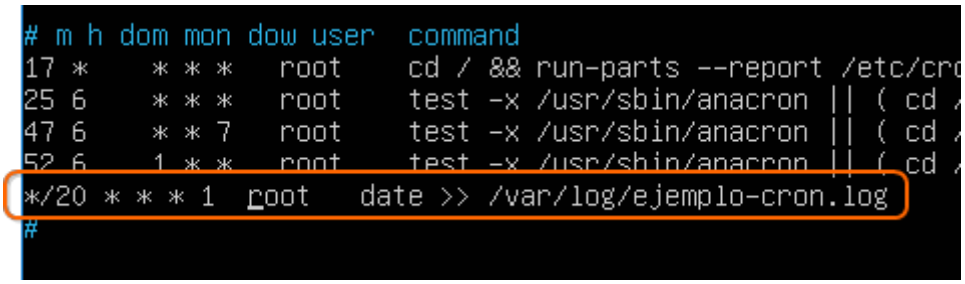
1.6 ESCRIBE UNA TAREA DE CRON

1.6.1 Se ejecuta únicamente los lunes cada 20 minutos

Para ejecutar una repetición de tiempo, utilizaremos la barra (*/20) y ya que se ejecutará los lunes, utilizaremos el 1, lo explico todo con más detalle en el siguiente punto.

1.6.2 Guarda un histórico con la fecha y hora del sistema de cada ejecución en el fichero /var/log/ejemplo-cron.log

Abrimos con sudo, el fichero /etc/crontab y añadimos la siguiente línea:



Vamos a analizar la línea:

Cada número tiene una función:

Minutos (m), Horas (h), Día del mes (dom), Mes (mon), Día de la semana (dow), Usuario (user), Comando (command)

*/20 → Cada 20 minutos

* → No se especifica hora

* → No se especifica día del mes

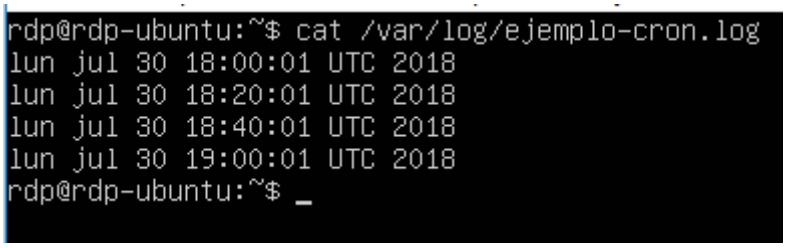
* → No se especifica el mes

1 → Representa al lunes

root → Usuario que va a ejecutar el comando (el directorio /var/log/ solo root puede modificarlo, por defecto, claro está).

date >> /var/log/ejemplo-cron.log → Comando que guardará un histórico con la hora y fecha del sistema cada vez que la tarea sea ejecutada.

Nos esperamos una hora y visualizamos el fichero /var/log/ejemplo-cron.log:

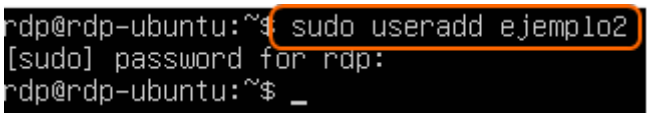


Como podemos observar, cada 20 min se ha ejecutado la tarea, ya que de casualidad, estamos a lunes.

1.7 COMANDO PARA CREAR UN USUARIO

1.7.1 Llamado ejemplo2

Para crear el usuario empezamos con el siguiente comando:



1.7.2 Su home debe estar en /var/www/html/ejemplo2

Lo primero de todo es crear el directorio indicado:

```
sudo mkdir /var/www/html/ejemplo2
```

Le asignamos como dueño a nuestro usuario ejemplo2 que creamos anteriormente:

```
sudo chown -R ejemplo2:ejemplo2 /var/www/html/ejemplo2
```

Comprobamos los permisos y si se ha realizado correctamente:

```
drwxr-xr-x 2 ejemplo2 ejemplo2  4096 jul 30 19:15 ejemplo2
```

Tal y como vemos, el directorio pertenece a ejemplo2 y los permisos, gracias a la máscara por defecto, tiene el 755. Para el grupo y otros, solo lectura y ejecución (101).

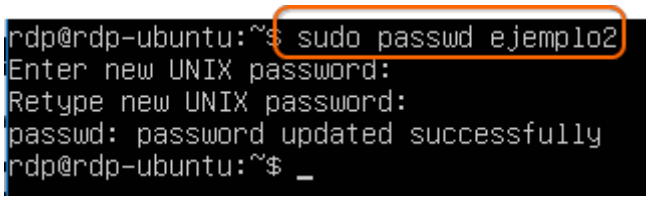
Ahora el comando más importante, vamos a asignarle el directorio, al home del usuario ejemplo2:

```
sudo usermod -d /var/www/html/ejemplo2 ejemplo2
```

El argumento –d sirve para indicar el directorio home del usuario, que éste está indicado en la última parte del comando. Cuando le establezcamos contraseña en el siguiente punto, haremos login y veremos que aún hay que resolver algunos problemas.

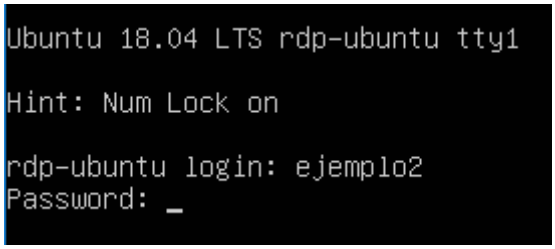
1.8 ESCRIBE EL COMANDO PARA CAMBIAR LA CONTRASEÑA DEL USUARIO EJEMPLO2

Para cambiar la clave de nuestro usuario, lo haremos con el siguiente comando:



Nos pedirá dos veces la clave para nuestro usuario.

Comprobamos que los cambios se han realizado correctamente:

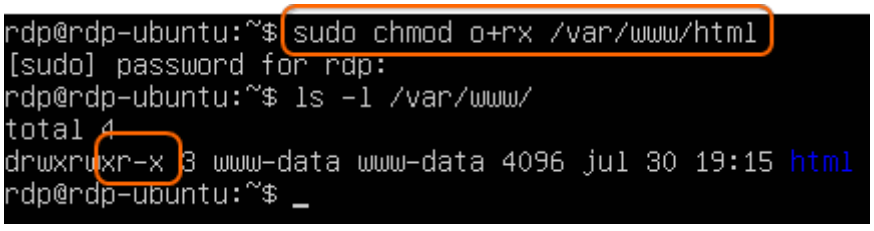


El login está perfecto con la contraseña que le hemos establecido, pero me he encontrado con un problema, al iniciar sesión me dice que no tiene directorio home asignado:

Este problema, que en este caso me ha pasado a mí, es debido a los permisos del directorio html:

Los permisos son 770 para el usuario dueño, que es www-data, es decir, para otros usuarios, no pueden ni leer lo que hay dentro.

Vamos a darle permiso de lectura y ejecución a otros usuarios para que el usuario ejemplo2 pueda al menos acceder a su directorio:



La parte del comando **o+rx** significa que para otros (other: o) le damos (+) lectura (r) y ejecución (x) al directorio /var/www/html.

Volvemos a hacer login con el usuario ejemplo2:

```
$ pwd
/var/www/html/ejemplo2
$ _
```

Al poner el comando pwd comprobamos en el directorio que estamos, nada más hacer login, y como podemos observar en la imagen, el usuario ejemplo2 está en su directorio home.

Otro detalle que le falta al usuario, es asignarle un Shell, para ello usaremos este comando:

```
rdp@rdp-ubuntu:~$ sudo usermod -s /bin/bash ejemplo2
[sudo] password for rdp:
rdp@rdp-ubuntu:~$
```

El comando usermod ya lo conocemos, esta vez le hemos establecido el argumento –s para indicarle el tipo de Shell que vamos a utilizar, en este caso **/bin/bash.**

Te preguntarás, ¿qué es un Shell? Es un intérprete de comandos.

Comprobamos que funciona volviendo a hacer login con el usuario ejemplo2:

```
ejemplo2@rdp-ubuntu:~$ pwd
/var/www/html/ejemplo2
ejemplo2@rdp-ubuntu:~$ _
```

Como vemos, ya tenemos en el intérprete, el **usuario@equipo** que estamos usando. En este caso es ejemplo2 como usuario y rdp-ubuntu como nombre de equipo de mi máquina virtual.

1.9 COMANDO PARA CONECTARNOS POS SSH

El comando es el siguiente:

```
ssh -p 2022 ubuntu@prueba.ejemplo.com
```

En los tres puntos siguientes, vamos a indicar cada una de las partes:

1.9.1 Al servidor prueba.ejemplo.com

```
ssh -p 2022 ubuntu@prueba.ejemplo.com
```

1.9.2 Con el usuario ubuntu

```
ssh -p 2022 ubuntu@prueba.ejemplo.com
```

1.9.3 Puerto 2022

```
ssh -p 2022 ubuntu@prueba.ejemplo.com
```

Analizando el comando y sus partes:

ssh –p (puerto) 2022 (número de puerto) usuario@host

1.10 AÑADE UNA VARIABLE DE ENTORNO LLAMADA PRUEBA CUYO VALOR SEA NOSOLOSOFTWARE PARA LA CONSOLA ACTUAL

Utilizamos el siguiente comando:

```
ejemplo2@rdp-ubuntu:~$ export PRUEBA="nosolosoftware"
```

El comando export, seguido del nombre que queremos darle a la variable, en este caso PRUEBA, y seguimos con el valor: =”nosolosoftware”, que lo pongo entre comillas ya que es una cadena de texto.

Ahora vamos a mostrar el contenido de la variable:

```
ejemplo2@rdp-ubuntu:~$ echo $PRUEBA
nosolosoftware
```

Para usar una variable, debemos añadirle el símbolo del dólar ($) delante del nombre de la variable, en este caso $PRUEBA, anteponiéndole el comando echo que es para mostrar el contenido de dicha variable, echo $PRUEBA.

Con el comando **env**, podemos ver las variables de entorno que tenemos creadas:

```
ejemplo2@rdp-ubuntu:~$ env
LANG=es_ES.UTF-8
INVOCATION_ID=741c6e9d716945b58c48fc245d903378
PRUEBA=nosolosoftware
XDG_VTNR=1
XDG_SESSION_ID=35
HUSHLOGIN=FALSE
USER=ejemplo2
PWD=/var/www/html/ejemplo2
HOME=/var/www/html/ejemplo2
JOURNAL_STREAM=9:31774
XDG_DATA_DIRS=/usr/local/share:/usr/share:/var/lib/snapd/desktop
MAIL=/var/mail/ejemplo2
SHELL=/bin/bash
TERM=linux
SHLVL=1
XDG_SEAT=seat0
LOGNAME=ejemplo2
XDG_RUNTIME_DIR=/run/user/1002
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/u
bin
_=/usr/bin/env
ejemplo2@rdp-ubuntu:~$
```

1.11 COMANDO PARA VER LOS REGISTROS MX DE GOOGLE.ES

Usaremos el siguiente comando:

```
ejemplo2@rdp-ubuntu:~$ dig MX google.es

; <<>> DiG 9.11.3-1ubuntu1.1-Ubuntu <<>> MX google.es
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 4129
;; flags: qr rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;google.es.                     IN      MX

;; ANSWER SECTION:
google.es.              600     IN      MX      10 aspmx.l.google.com.
google.es.              600     IN      MX      40 alt3.aspmx.l.google.com.
google.es.              600     IN      MX      20 alt1.aspmx.l.google.com.
google.es.              600     IN      MX      30 alt2.aspmx.l.google.com.
google.es.              600     IN      MX      50 alt4.aspmx.l.google.com.

;; Query time: 39 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Tue Jul 31 12:00:40 UTC 2018
;; MSG SIZE  rcvd: 156

ejemplo2@rdp-ubuntu:~$ _
```

Con el comando **dig** seguido del tipo de registro dns que queremos consultar, que en este caso es **MX**, seguido del dominio, que en este caso es **google.es**.

Esta consulta nos la ha realizado nuestro propio servidor dns.

Si quisiéramos que otro servidor de nombres (NS), en este caso el de google, nos lo facilite, solo tendríamos que indicarlo añadiendo antes del dominio, la ip de dicho servidor de nombres, precedida de un arroba (@):

```
ejemplo2@rdp-ubuntu:~$ dig MX @8.8.8.8 google.es

; <<>> DiG 9.11.3-1ubuntu1.1-Ubuntu <<>> MX @8.8.8.8 google.es
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 26441
;; flags: qr rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;google.es.                     IN      MX

;; ANSWER SECTION:
google.es.              599     IN      MX      40 alt3.aspmx.l.google.com.
google.es.              599     IN      MX      50 alt4.aspmx.l.google.com.
google.es.              599     IN      MX      10 aspmx.l.google.com.
google.es.              599     IN      MX      30 alt2.aspmx.l.google.com.
google.es.              599     IN      MX      20 alt1.aspmx.l.google.com.

;; Query time: 37 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Tue Jul 31 12:04:23 UTC 2018
;; MSG SIZE  rcvd: 156
```

1.12 ¿QUÉ HACE EL COMANDO FIND /HOME/UBUNTU | GREP –I ID_RSA?

El comando busca en el directorio **/home/ubuntu,** filtrando el resultado con el comando **grep** y el argumento **–i** que sirve para ignorar mayúsculas y minúsculas, buscando un directorio o archivo con el nombre **id_rsa**.