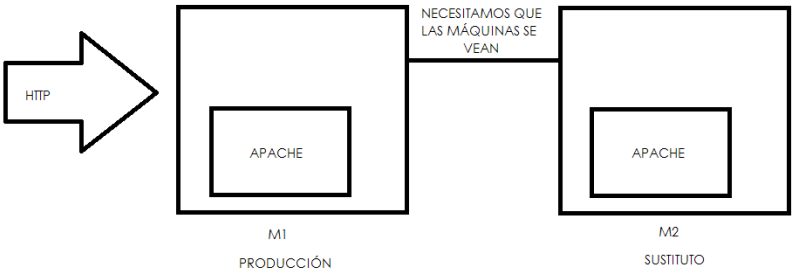
PRÁCTICA 2

Posibles opciones para resolver el problema:



1. Copiar información usando el comando: SCP
scp ipm1 /tmp/f.txt /tmp/
(Cogería el archivo de M1 y lo guardaría en la carpeta /tmp/ o dónde quisiéramos)
Justo después de esto, el sistema nos pedirá la contraseña del usuario, esto podría ser un gran problema porque cada vez que actualicemos, tendríamos que disponer de una persona física que se encargase de escribir la contraseña todo el tiempo.

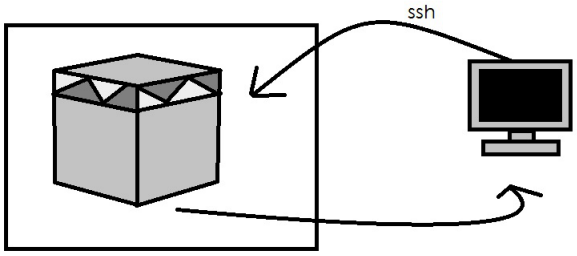
2. Esta es otra orden más compleja:
tar czf - /var/www/ ssh ipm2 'cat> /tmp/web.tgz'
La primera parte lo que haría sería comprimir lo que hay en el espacio web y luego lo enviaría.
A continuación nos pedirá la contraseña de la máquina 1.
Tenemos que tener cuidado si ejecutamos esta orden muchas veces, ya que se podrían solapar los procesos, por tanto no es viable en el caso de que tengamos que transferir muchos Gbs.

Para optimizar este problema, vamos a usar otro método para que en las actualizaciones de las máquinas sólo **se transfieran la parte que cambiamos** de un sitio a otro. Que **no nos pidan mas veces contraseñas** al enviar archivos y además se haga de una **manera programada.**

Resolución de la práctica 2:

Usando RSYNC(sólo se copia lo modificado)
La usamos sobre la máquina2, la máquina2 miraría las diferencias de sus directorios seleccionados con la máquina1 y pediría los cambios que habrían desde entonces.
Para ello usamos la siguiente orden:
rsync -avz – e ssh ipm1:/var/www/ /var/www/
origen espacio web propio

Justo después de esto tenemos que añadir la contraseña de nuevo.
Para evitar que nos pida la contraseña de nuevo:



Conseguir que **ssh** funcione sin clave:

Forma remota a través del terminal

(Hacer que las máquinas confíen, para no necesitar autorización)
Usaremos dos herramientas y dos comandos:

```
swap1@swap1:/var/www$ ssh-keygen -b 4096 -t rsa_
```

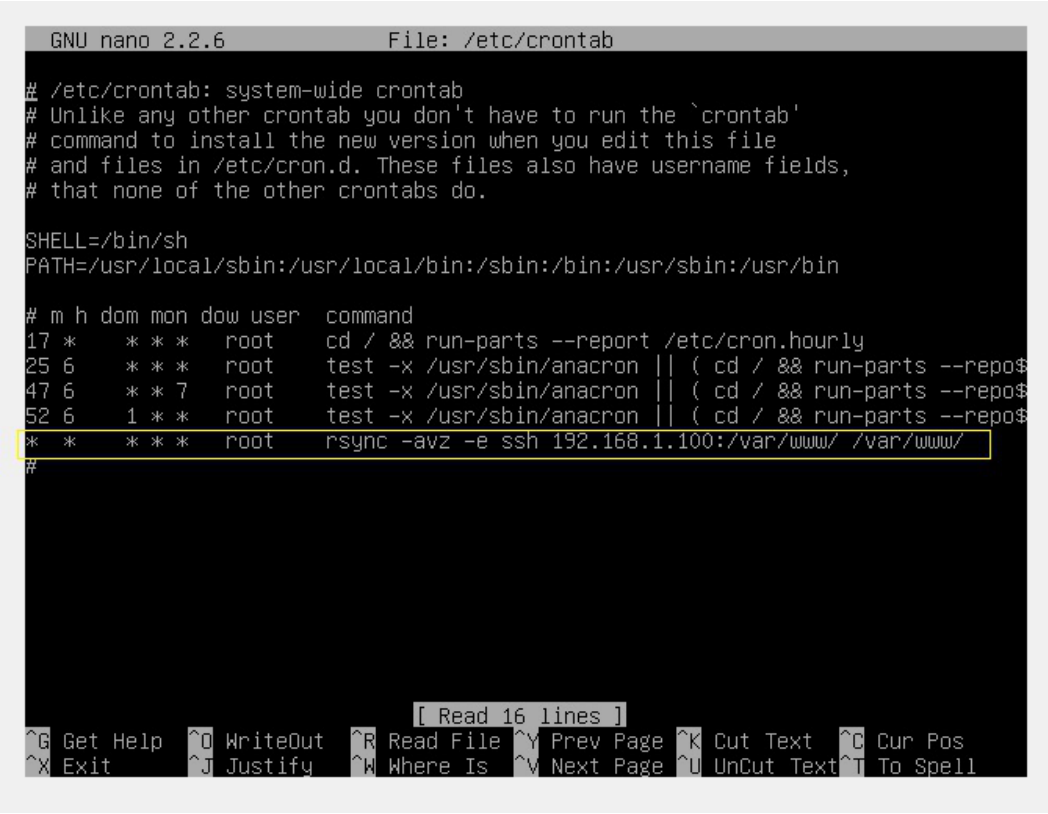
ssh-copy-id -i ~/.ssh/id_dsa.pub (fichero para acceder sin contraseña) ipmaq1

Las claves se copian y ya no necesitamos volver a escribirlas de nuevo:

```
swap1@swap1:/var/www$ ssh-copy-id 192.168.1.100_
```

Si todo va bien con: ssh ipmaq1 obtendríamos acceso a ella: $_

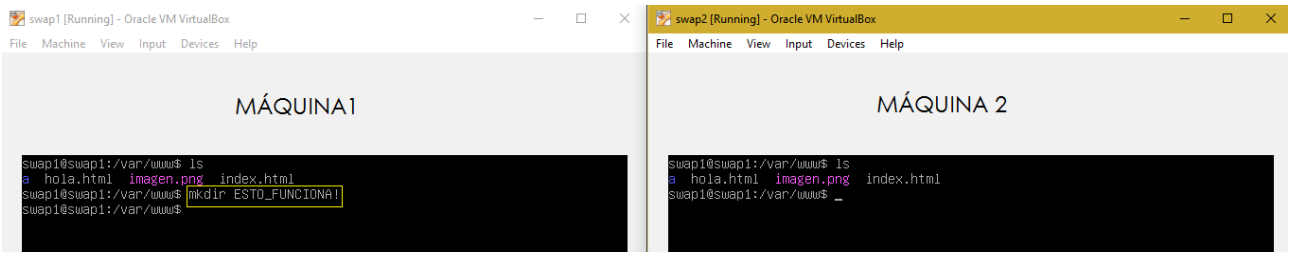
Ahora tenemos que configurar una tarea de manera automática:
Usando ***Crontab*** `swap1@swap1:/var/www$ sudo nano /etc/crontab`



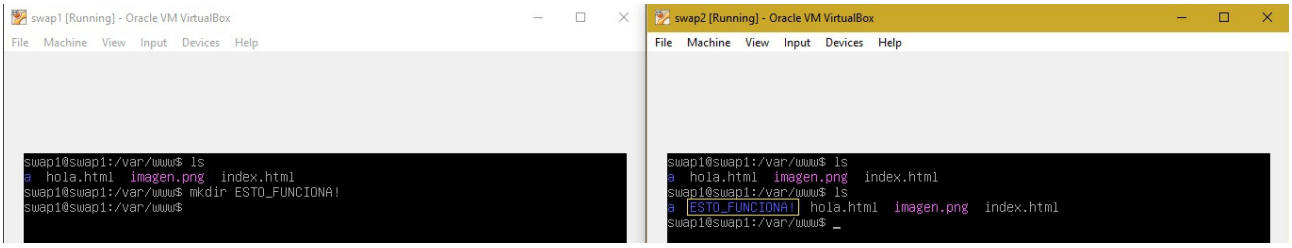
Ejemplos de formato:

```
30 0 * * * una vez al día a las 00:30
10 * * * * todas las horas con minuto 10
* * * * *  cada minuto
*/3 * * * * cada 3 minutos
*/5* * * * cada 5minutos
35  11* * * una vez al día a las 11:35
45 * * * * todas las horas con minuto 45
```

Seleccionamos el que sea más apropiado para nuestro problema y así resolveríamos el ejercicio propuesto:



Creamos un nuevo directorio en la máquina 1 y tras esperar el tiempo definido con Crontab,



vemos que éste aparece en la máquina 2: