Cómo realizar copias de seguridad con rsync

Enlace original al documento

Una de las tareas más importantes de un webmaster, y muchas veces olvidada, es la realización de copias de seguridad. Hay multitud de aplicaciones comerciales y open source que nos permiten realizarlas de una forma más o menos potente o sencilla. En este artículo vamos a explicar mediante un ejemplo práctico como obtener una réplica local de la información de un servidor mediante el comando rsync. Para ello antes de nada vamos a dar una breve introducción a este comando y sus características fundamentales, las cuales nos permitirán montar el sistema de réplica en muy pocos pasos y de forma sencilla. Dejamos al interés del usuario el ir más allá y montar sistemas más potentes aprovechando todas las posibilidades existentes. Introduction a rsync

La utilidad rsync, escrita inicialmente por Andrew Tridgell y Paul Mackerras, viene prácticamente con la totalidad de distribuciones Linux y sistemas Unix. En caso de no ser así se dispone del código fuente en rsync.samba.org. La principal utilidad de rsync es la de sincronizar estructuras de árboles de directorios a través de la red, aunque puede ser utilizado perfectamente también dentro de una máquina de forma local.

Es muy fácil de utilizar y configurar, y al contrario que la utilización de programas de script basados en FTP, ofrece una serie de funcionalidades que lo diferencian claramente. El algoritmo que utiliza envía únicamente la información que ha cambiado dentro de cada archivo, evitando enviar el archivo completo, y permite comprimirla para reducir la utilización de ancho de banda, o enviarlo a través de ssh si se requiere un nivel extra de seguridad en la transmisión.

Hay diferentes maneras de utilizar rsync en función de lo que necesitemos. Permite realizar copias locales, copias desde y hasta un servidor que corra un demonio de rsync, copias desde y hasta un servidor utilizando un shell remoto como transporte, y por último nos permite listar archivos de un servidor remoto. Todas estas características nos permiten realizar multitud de posibilidades como copias locales, copias de servidores remotos, sistemas de mirroring, mantenimiento sincronizado de sistemas de preproducción y producción, etc…

Como ya hemos visto la utilización de rsync nos permite realizar transmisiones ecónomicas en cuanto a ancho de banda, debido a su algoritmo basado en las diferencias de los archivos, y en la utilización de la compresión, pero ¿qué hay acerca de la seguridad?. El protocolo de auntenticación que se usa está basado en un MD4 de 128 bits. Este sistema se considera suficientemente bueno para cualquier necesidad, aunque para aquellos más paranoicos existe la opción de utilizar rsync sobre ssh, o incluso mezclar la flexibilidad de ejecutar un demonio rsync utilizando ssh como transporte. Hay que tener en cuenta que rsync no realiza ningún tipo de encriptación de la información transmitida, únicamente se utiliza en el momento de la autenticación. Qué es lo que queremos hacer

El ejemplo que vamos a desarrollar está basado en la utilización de un servidor remoto que corre un demonio de rsync, que exporta diferentes árboles de directorios, llámados módulos. Vamos a utilizar un planteamiento típico utilizando un servidor remoto rsync, pero sin utilizar ssh como transporte, ya que no se considera que la información a transmitir requiera ese nivel de seguridad. Utilizaremos también un equipo local en el que queremos obtener una réplica de lo que hay en los módulos exportados por el servidor de rsync.

Los backups que se realizan de esta forma son realmente una réplica de lo que hay en el servidor, por lo que si tenemos algún problema bastará con subir los archivos o carpetas directamente.

Estas son las carpetas del servidor de las que queremos hacer copias:

- /www ⇒ los webs de los clientes
- /var/lib/mysql ⇒ bases de datos de mysql

Esta es la carpeta del equipo local donde queremos realizar las copias:
/usr/local/backups/replica_hosting Configuración del servidor

Vamos a ejecutar el demonio de rsync a partir del superdemonio xinetd, para lo cual debemos asegurarnos de que está asociado al puerto 873/tcp dentro del archivo /etc/services. Para ello hay que comprobar que el archivo contiene un línea como la siguiente: rsync 873/tcp

Una vez hecho esto debemos activar el servicio dentro de xidentd. Si existe el archivo /etc/xinetd.d/rsync bastará con activarlo poniendo la entrada disable a no, en caso de que el archivo no exista se puede crear con el siguiente contenido

service rsync {

```
      disable = no
      socket_type     = stream
      wait            = no
      user            = root
      server          = /usr/bin/rsync
      server_args     = --daemon
      log_on_failure  += USERID
```

}

Con esto ya tendríamos preparado el servidor de rsync para escuchar las peticiones entrantes, únicamente nos queda configurar el servicio y relanzar xinetd para que comience a trabajar.

Para preparar la configuración del servidor debemos crear dos archivos, /etc/rsyncd.conf y /etc/rsyncd.secrets. El primero de ellos contiene la configuración de los módulos exportados, mientras que el segundo contiene la información de los usuarios que tienen permisos para realizar conexiones. Estos son los contenidos de cada archivo:

/etc/rsyncd.conf

uid = nobody gid = nobody use chroot = yes max connections = 1 log file = /var/log/rsyncd.log pid file = /var/run/rsyncd.pid

[www] path = /www comment = datos de www use chroot = true max connections = 1 read only = true list = false uid = nobody gid = nobody auth users = remote_user secrets file = /etc/rsyncd.secrets strict modes = true hosts allow = 111.111.111.111

[mysql] path = /var/lib/mysql comment = datos de mysql use chroot = true max connections = 1 read only = true list = false uid = mysql gid = mysql auth users = remote_user secrets file = /etc/rsyncd.secrets strict modes = true hosts allow = 111.111.111.111

/etc/rsyncd.secrets

remote_user:thepass

El archivo rsyncd.conf se puede observar que se compone de una primera parte en la que se configuran parámetros genéricos del servidor, y una segunda parte donde se muestra la configuración exacta de cada módulo exportado. En la configuración se ha puesto por defecto que se use como usuario local nobody, y que se haga bajo un entorno chroot por seguridad. Esto hace que por defecto cuando nos conectemos a un módulo lo hagamos con ese usuario, y no podamos salir de él a otras zonas del servidor. Dentro de cada módulo se indica la carpeta a la que apunta, se dice que únicamente se pueden leer archivos, mediante read only = true, pero no modificarlos ni crear nuevos. Se dice también que si se socilita un listado de módulos al servidor estos estén ocultos, que se utilice como datos de conexión un usuario de nombre remote_user cuyo password está en el archivo /etc/rsyncd.secrets, y que únicamente se permitan conexiones desde la ip 111.111.111.111

En la configuración del segundo módulo se puede ver como se ha cambiado el usuario por defecto de nobody a mysql, con el fin de poder realizar las copias de las bases de datos, ya que normalmente el usuario nobody no tendrá permiso para leerlas.

Estos deberían ser lor permisos para ambos archivos:

-rw-r–r– 1 root root 472 Dec 31 14:01 /etc/rsyncd.conf -rwx—— 1 root root 21 Dec 31 14:00 /etc/rsyncd.secrets

Hay multitudes de opciones de configuración que nos permiten configurar el servidor para nuestras necesidades específicas, más información en la página de man de rsyncd.conf.

Una vez configurado el servidor, deberíamos reiniciar xinetd con el fin de poder conectarnos al servidor de rsync

bash# /etc/rc.d/init.d/xinetd restart

Ahora ya podemos comenzar a configurar el equipo cliente. Configuración del Cliente

Dentro de la carpeta /usr/local/backups creamos el archivo password.rsync, en el que escribimos el password con el que queremos conectarnos al servidor rsync: thepass

Debemos darle permisos de sólo lectura para el usuario que vaya a ejecutar la copia.

bash# chmod 600 password.rsync

Ahora ya podemos probar la conexión al servidor, para ello lo más sencillo es pedirle que nos dé un listado de lo exportable en uno de de los módulos, por ejemplo:

bash# rsync –password-file=/usr/local/backups/password.rsync rsync:*remote_user@servidor.web.com/www Con esta orden el servidor nos contestará con el conjunto de archivos y carpetas disponibles en el módulo www. Si quisiéramos que lo hiciera de manera recursiva, recorriendo las subcarpetas, deberíamos haberle pasado el parámetro -r. Hay que prestar atención a como acaba la información que pedimos al servidor. En este caso le estamos pidiendo www, que es diferente que ponerle una barra al final em>www/. Si realizamos una copia a local a partir de la primera opción se nos crearía una carpeta www y dentro el contenido de la misma, sin embargo con la segunda opción se copiaría directamente el contenido de la carpeta. Si el servicio no funciona disponemos de la opción de mirar el log del servidor /var/log/rsyncd.log para investigar las posibles causas del problema. Conviene asegurarse de que el servicio está disponible en xinetd reiniciando éste si es necesario, y que los datos de usuario son correctos. Una vez probado que funciona el servicio podemos preparar un pequeño script que ejecute la petición al servidor para cada módulo disponible. Este script podríamos luego incorporarlo a un cron, o modificarlo lo que hiciera*

*falta para montar copias totales semanales e incrementales diarias, hay información al respecto en las referencias al final del artículo. Creamos el archivo replica.sh (son tres líneas!!!!) #/bin/sh rsync -arzvl –password-file=/usr/local/backups/password.rsync rsync:*remote_user@servidor.web.com/www /usr/local/backups/replica_hosting rsync -arzvl –password-file=/usr/local/backups/password.rsync rsync:*remote_user@servidor.web.com/mysql /usr/local/backups/replica_hosting Dándole permisos de ejecución ya es suficiente para lanzar el script y obtener una copia de toda la información del servidor. Sucesivas ejecuciones del script producirán que se actualice la información existente en base al algoritmo de rsync, consiguiendo una replica actualizada con un consumo de ancho de banda óptimo. bash# chmod 755 replica.sh bash# ./replica.sh Más información * *http://samba.anu.edu.au/rsync/* * *http://everythinglinux.org/rsync/* * *http://www.mikerubel.org/computers/rsync_snapshots/* * *http://finmath.uchicago.edu/~wilder/Security/rsync/*