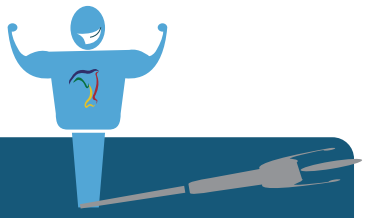
autentia

Avenida de Castilla,1 - Edificio Best Point - Oficina 21B
28830 San Fernando de Henares (Madrid)
tel./fax: +34 91 675 33 06
info@autentia.com - **www.autentia.com**

¿**Qué ofrece** Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
Ese apoyo que siempre quiso tener...

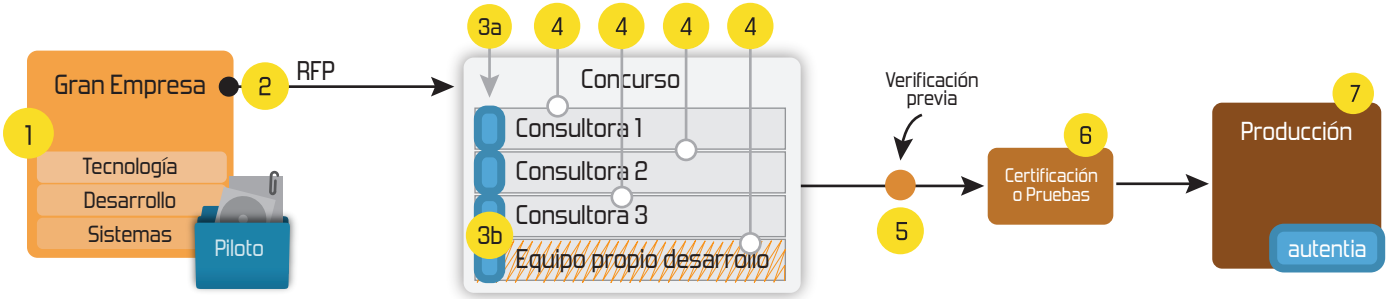
1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

- Spring MVC, JSF-PrimeFaces /RichFaces, HTML5, CSS3, JavaScript-jQuery
- Gestor portales (Liferay)
- Gestor de contenidos (Alfresco)
- Aplicaciones híbridas
- Control de autenticación y acceso (Spring Security)
- UDDI
- Web Services
- Rest Services
- Social SSO
- SSO (Cas)
- JPA-Hibernate, MyBatis
- Motor de búsqueda empresarial (Solr)
- ETL (Talend)
- Dirección de Proyectos Informáticos.
- Metodologías ágiles
- Patrones de diseño
- TDD
- Tareas programadas (Quartz)
- Gestor documental (Alfresco)
- Inversión de control (Spring)
- BPM (jBPM o Bonita)
- Generación de informes (JasperReport)
- ESB (Open ESB)

Compartimos nuestro conocimiento en:
www.adictosaltrabajo.com

Para más información visítenos en:
www.autentia.com



autentia
Soporte a desarrollo informático
Hosting patrocinado por
enredados

Entra en Adictos a través de
E-mail
Contraseña
Entrar
Deseo registrarme
Olvidé mi contraseña



Daniel Ventas López

Becario en autentia.

Ingeniero técnico en informática, especialidad en sistemas.

Puedes encontrarme en Autentia: Ofrecemos servicios de soporte a desarrollo, factoría y formación

Somos expertos en Java/J2EE

Ver todos los tutoriales del autor

Fecha de publicación del tutorial: 2014-01-03 Tutorial visitado 5 veces Descargar en PDF

Cómo montar sistema raid1 para replicación de datos en una debian funcionando.

Índice de contenidos

1. Introducción
2. Entorno
3. Configurando el entorno.
4. Conclusiones.

1. Introducción

En este tutorial vamos a ver una forma sencilla de replicar los datos que no queramos perder de un disco duro, haciendo un raid1 con un disco duro auxiliar del mismo tamaño del que está funcionando. Para ello nos ayudaremos de la potencia de las herramientas de linux, en este caso mdadm.

2. Entorno

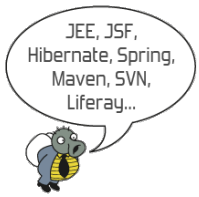
El tutorial está escrito usando el siguiente entorno:

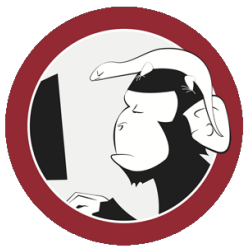
- Hardware: Portátil MacBook Pro 17' (2,93 GHz Intel Core 2 Duo, 8GB 1067 Mhz DDR3, 320GB Flash Storage).
- VirtualBox-4.3.6
- Sistema Operativo: Debian wheezy 7.3
- mdadm - v3.2.5

3. Configurando el entorno.

Para el ejemplo vamos tener sobre una máquina virtual una debian, ya funcionando, a la que le vamos a incluir un disco duro del mismo tamaño. Lo dejamos como vemos en la imagen.

Catálogo de servicios Autentia





Síguenos a través de:

Últimas Noticias

» IX Autentia Cycling Day (ACTUALIZADO)

» Autentia en la carrera de las empresas

» Spring 4.0 ¿qué hay de nuevo amigo?

» Torneo de pádel solidario AMEB

» Próxima charla: Gradle como alternativa a Maven para la construcción de proyectos en Java

Histórico de noticias

Últimos Tutoriales

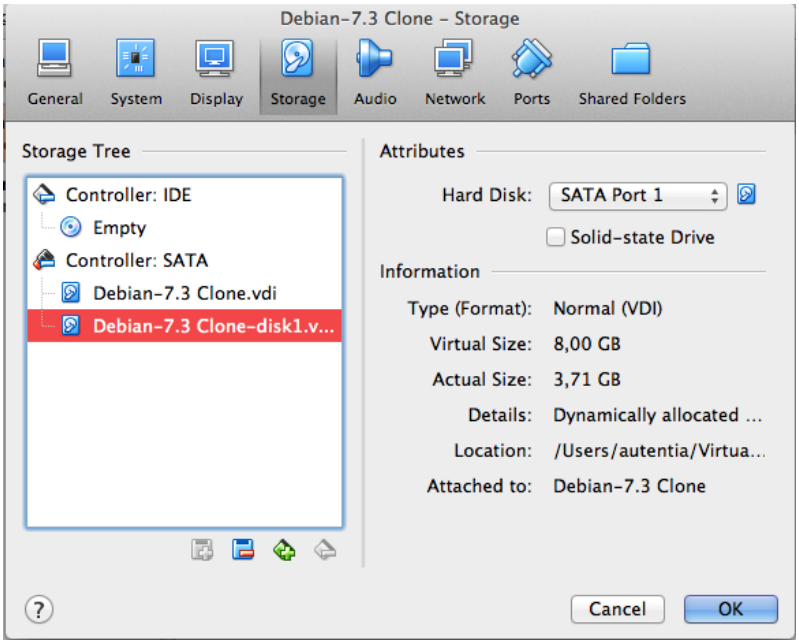
» Notificaciones locales en iOS.

» Configuración básica de seguridad en servidor linux con iptables y fail2ban

» Desarrollo rápido de aplicaciones CRUD con OpenXava

» ApacheDS: tests de integración contra un servidor LDAP embebido.

» ¿Mockear métodos estáticos?, con el soporte de



Ahora arrancamos la máquina y nos dejamos de entorno gráfico, vamos al querido terminal.

Lo primero es instalar la herramienta que nos va a ayudar a hacer el raid, con el siguiente comando:

```
# apt-get install mdadm
```

Vamos a copiar la estructura del disco de origen (sda) al disco de destino (sdb) con el siguiente comando:

```
# sfdisk --dump /dev/sda | sfdisk --force /dev/sdb
```

Y nuestra estructura de discos es la siguiente:

```
root@debian:~# fdisk -l

Disk /dev/sda: 8589 MB, 8589934592 bytes
255 heads, 63 sectors/track, 1044 cylinders, total 16777216 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000928f3

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *        2048      684031      340992   83  Linux
/dev/sda2          686078    16775167     8044545    5  Extended
/dev/sda5          686080     6537215     2925568   83  Linux
/dev/sda6         6539264     9408511     1434624   83  Linux
/dev/sda7         9410560    10356735      473088   82  Linux swap / Solaris
/dev/sda8        10358784    10852351      246784   83  Linux
/dev/sda9        10854400    16775167     2960384   83  Linux

Disk /dev/sdb: 8589 MB, 8589934592 bytes
255 heads, 63 sectors/track, 1044 cylinders, total 16777216 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1   *        2048      684031      340992   83  Linux
/dev/sdb2          686078    16775167     8044545    5  Extended
/dev/sdb5          686080     6537215     2925568   83  Linux
/dev/sdb6         6539264     9408511     1434624   83  Linux
/dev/sdb7         9410560    10356735      473088   82  Linux swap / Solaris
/dev/sdb8        10358784    10852351      246784   83  Linux
/dev/sdb9        10854400    16775167     2960384   83  Linux
```

- **sda1**: está la partición /
- **sda5**: está la partición /usr
- **sda6**: está la partición /var
- **sda7**: está la partición swap
- **sda8**: está la partición /tmp
- **sda9**: está la partición /home

El resultado que queremos es el siguiente:

```
/dev/md0 (sda1 + sdb1) mounted on /
/dev/md5 (sda5 + sdb5) mounted on /usr
/dev/md6 (sda6 + sdb6) mounted on /var
```

PowerMock.

Últimos Tutoriales del Autor

» Configuración básica de seguridad en servidor linux con iptables y fail2ban

» Cómo integrar un Job de Talend a nuestro proyecto Java

» Minimizar código con anotaciones en Spring.

Últimas ofertas de empleo

2011-09-08
Comercial - Ventas - MADRID.

2011-09-03
Comercial - Ventas - VALENCIA.

2011-08-19
Comercial - Compras - ALICANTE.

2011-07-12
Otras Sin catalogar - MADRID.

2011-07-06
Otras Sin catalogar - LUGO.

```
/dev/md7 (sda7 + sdb7) used for swap
/dev/md8 (sda8 + sdb8) mounted on /tmp
/dev/md9 (sda9 + sdb9) mounted on /home
```

El siguiente paso es cambiar el tipo de partición del sdb para que coincidan con fd Linux raid autodetect. Para ello nos ayudamos de la herramienta fdisk.

```
# fdisk /dev/sdb
```

En el menú que aparece pulsamos '**t**' para cambiar el id de la partición, nos preguntará qué partición y pulsamos '**1**', nos volverá a preguntar el código hexadecimal del sistema que le queremos asignar y pulsamod '**fd**'. Y así con cada una de las particiones que tiene el disco sdb y queramos poner en raid. Cuando terminemos pulsamos sobre la tecla '**w**' para que escriba los cambios.

Ahora vemos los cambios con fdisk :

```
Disk /dev/sdb: 8589 MB, 8589934592 bytes
255 heads, 63 sectors/track, 1044 cylinders, total 16777216 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1   *        2048      684031      340992   fd  Linux raid autodetect
/dev/sdb2          686078    16775167     8044545    5  Extended
/dev/sdb5          686080     6537215     2925568   fd  Linux raid autodetect
/dev/sdb6         6539264     9408511     1434624   fd  Linux raid autodetect
/dev/sdb7         9410560    10356735      473088   fd  Linux raid autodetect
/dev/sdb8        10358784    10852351      246784   fd  Linux raid autodetect
/dev/sdb9        10854400    16775167     2960384   fd  Linux raid autodetect
```

Ahora vamos a crearnos la interfaz que sustituirá las particiones con la herramienta mdadm.

```
# mdadm --create /dev/md1 --chunk=64 --level=1 --raid-devices=2 missing /dev/sdb1
# mdadm --create /dev/md5 --chunk=64 --level=1 --raid-devices=2 missing /dev/sdb5
# mdadm --create /dev/md6 --chunk=64 --level=1 --raid-devices=2 missing /dev/sdb6
# mdadm --create /dev/md7 --chunk=64 --level=1 --raid-devices=2 missing /dev/sdb7
# mdadm --create /dev/md8 --chunk=64 --level=1 --raid-devices=2 missing /dev/sdb8
# mdadm --create /dev/md9 --chunk=64 --level=1 --raid-devices=2 missing /dev/sdb9
```

Para comprobar que se han creado correctamente deberíamos de ver lo siguiente:

```
# cat /proc/mdstat
```

```
root@debian:~# cat /proc/mdstat
Personalities : [raid1]
md1 : active (auto-read-only) raid1 sdb1[1]
      340672 blocks super 1.2 [2/1] [_U]

md5 : active (auto-read-only) raid1 sdb5[1]
      2923456 blocks super 1.2 [2/1] [_U]

md6 : active (auto-read-only) raid1 sdb6[1]
      1433536 blocks super 1.2 [2/1] [_U]

md7 : active (auto-read-only) raid1 sdb7[1]
      472768 blocks super 1.2 [2/1] [_U]
        resync=PENDING

md8 : active (auto-read-only) raid1 sdb8[1]
      246592 blocks super 1.2 [2/1] [_U]

md9 : active (auto-read-only) raid1 sdb9[1]
      2958272 blocks super 1.2 [2/1] [_U]

unused devices: <none>
```

Seguimos transformando nuestro nuevo disco para adaptarlo al raid, así que vamos a hacer 2 pasos por cada partición:

Primero damos formato a la partición así:

```
# mkfs.ext3 /dev/md1
```

Ahora copiamos desde el origen:

```
# dd if=/dev/sda1 of=/dev/sdb1
```

Y ahora realizamos la misma operación con md5, md6 y md9

Para md8 hay que dar permisos, para ello damos formato, montamos el dispositivo y damos permisos con las siguientes instrucciones:

```
# mkfs.ext3 /dev/md4
# mkdir /mnt/tmp
# mount /dev/md8 /mnt/tmp
# chmod 777 /mnt/tmp
```

Y por último nos falta la partición de swap que nos vale con la siguiente instrucción:

```
# mkswap /dev/md7
```

Bien ya tenemos todo listo para cambiar nuestro fstab e iniciar desde el raid, aunque de momento funciona como si estuviera sólo el disco duro nuevo, luego añadiremos el disco original al RAID.

Nuestro fichero /etc/fstab tiene que quedar como el mío:

```
root@debian:~# cat /etc/fstab
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
# / was on /dev/sda1 during installation
/dev/mod1   /   ext4    errors=remount-ro   0   1
# /home was on /dev/sda9 during installation
/dev/md9    /home   ext4    defaults    0   2
# /tmp was on /dev/sda8 during installation
/dev/md8    /tmp    ext4    defaults    0   2
# /usr was on /dev/sda5 during installation
/dev/md5    /usr    ext4    defaults    0   2
# /var was on /dev/sda6 during installation
/dev/md6    /var    ext4    defaults    0   2
# swap was on /dev/sda7 during installation
/dev/md7    none    swap    sw  0   0
/dev/sr0        /media/cdrom0   udf,iso9660 user,noauto     0       0
```

Bien reiniciamos y debería de aparecer en los logs del inicio que se carga todo bien, y nos aseguramos al reinicio con:

```
# cat /proc/mdstat
```

```
root@debian:~# cat /proc/mdstat
Personalities : [raid1]
md1 : active (auto-read-only) raid1 sdb1[1]
      340672 blocks super 1.2 [2/1] [_U]

md5 : active (auto-read-only) raid1 sdb5[1]
      2923456 blocks super 1.2 [2/1] [_U]

md6 : active (auto-read-only) raid1 sdb6[1]
      1433536 blocks super 1.2 [2/1] [_U]

md7 : active (auto-read-only) raid1 sdb7[1]
      472768 blocks super 1.2 [2/1] [_U]
        resync=PENDING

md8 : active (auto-read-only) raid1 sdb8[1]
      246592 blocks super 1.2 [2/1] [_U]

md9 : active (auto-read-only) raid1 sdb9[1]
      2958272 blocks super 1.2 [2/1] [_U]

unused devices: <none>
```

Por último queda añadir el disco original sda, para ello tenemos que configurar Linux raid autodetect como en el disco sdb, para dejarlo igual, como hemos explicado arriba. Y añadimos las particiones con el siguiente comando:

```
# mdadm --add /dev/md1 /dev/sda1
# mdadm --add /dev/md5 /dev/sda5
# mdadm --add /dev/md6 /dev/sda6
# mdadm --add /dev/md7 /dev/sda7
# mdadm --add /dev/md8 /dev/sda8
# mdadm --add /dev/md9 /dev/sda9
```

Bien si todo está correcto, tenemos que tener la siguiente salida:

```
root@debian:~# cat /proc/mdstat
Personalities : [raid1]
md1 : active (auto-read-only) raid1 sda1[2] sdb1[1]
      340672 blocks super 1.2 [2/2] [UU]

md5 : active raid1 sda5[2] sdb5[1]
      2923456 blocks super 1.2 [2/2] [UUU]

md6 : active raid1 sda6[2] sdb6[1]
      1433536 blocks super 1.2 [2/2] [UU]
```

```
md7 : active (auto-read-only) raid1 sda7[2] sdb7[1]
      472768 blocks super 1.2 [2/2] [UU]
        resync=PENDING

md8 : active raid1 sda8[2] sdb8[1]
      246592 blocks super 1.2 [2/2] [UU]

md9 : active raid1 sda9[2] sdb9[1]
      2958272 blocks super 1.2 [2/2] [UU]

unused devices: <none>
```

Si hacéis el comando inmediatamente después de añadir el disco sda veréis cómo van cargando los datos, cuando se complete debería quedar como os acabo de mostrar y podréis reiniciar, hasta entonces hay que esperar un poquito.

4. Conclusiones

Hemos visto una forma fácil de tener un backup automático a un disco duro conservando todos los datos y aunque falle un disco se podría seguir con el otro sin problemas.

Cualquier duda o aclaración en los comentarios.

Un saludo

A continuación puedes evaluarlo:

Regístrate para evaluarlo

Por favor, vota +1 o compártelo si te pareció interesante

Share |

Anímate y coméntanos lo que pienses sobre este **TUTORIAL:**

» **Registrate** y accede a esta y otras ventajas «

SOME RIGHTS RESERVED Esta obra está licenciada bajo licencia Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5

PUSH THIS Page Pushers Community Help?

---- no clicks

0 people brought clicks to this page

+ + + + + + + +

powered by karmacracy

Copyright 2003-2014 © All Rights Reserved | Texto legal y condiciones de uso | Banners | Powered by Autentia | Contacto

W3C XHTML 1.0 W3C CSS XML RSS XML ATOM