

# Eskuliburua

ELKARBACKUP



## Aurkibidea

1.-Zer da ElkarBackup.....	3
2.-Instalazioa.....	5
2.1.-Guk prestatutako Irudia erabili.....	5
2.2.-Aurrez instalatuta duzun Debian batean paketea instalatzea.....	6
2.3.-Eguneraketak.....	8
3.-Zerbitzarira konektatzen.....	9
4.-Bezeroak eta Lanak.....	10
4.1.-Baina, zer dira bezeroak eta lanak ?.....	10
4.2.-GNU/Linux Bezeroak gehitzen.....	11
4.3.-Windows Bezeroak gehitzen.....	17
4.4.-Akatsak konpontzen.....	23
5.-Politikak.....	25
Programazioa.....	26
Arrasto zaharrak.....	28
Politika ezberdinen erabilera.....	28
6.-Scriptak.....	30
6.1.-Script berri bat.....	30
7.-Aplikazioaren konfigurazioa .....	33
7.1.-Parametroak kudeatu.....	33
SSH gakoak.....	33
MySQL zerbitzaria.....	33
Posta elektroniko bidezko mezuak.....	33
Kuota abisuak.....	34
Beste parametro batzuk.....	35
7.2.-Backup kokapena kudeatu.....	36
7.3.-Biltegiaren kopia.....	37
Automatizazioa.....	39
8.-Oinarrien errepasso txiki bat: Rsnapshot.....	43
8.1.-Maitasuna.....	43
8.2.-Erretentzioa.....	44
8.3.-Errotazioa.....	44
9.-Eranskinak.....	46
9.1.-Irudia deskargatu ondoren.....	46

---

9.2.-Errepositorioa konprimitzeko scripta.....	46
9.3.-Lanak ordenatu.....	48
9.4.-Irudia beste birtualizazio sistema batzuetara bihurtzen.....	49
9.5.-Hodeiarekin sinkronizazioa.....	49
9.6.-Windows bezeroen kopiak eta snapshot-ak.....	49
9.7.-NFS urruneko diskoa.....	50
10.-Lizentzia.....	51

## 1.- Zer da ElkarBackup

Ez da gure asmoa hemen segurtasun kopia sistema bat zer den azaltzea, ezta ere datuen segurtasuna bermatzea zein garrantzitsua den azpimarratzea, edo zergatik izan behar dugun kopia prozesu hau automatizatua. Hala ere, argi izan behar dugu:

- Murphy, gure aspaldiko lagun hori, gutxien espero dugunean agertu daitekeela bisita egitera, eta prestatuta egotea komeni dela
- Maiztasun ezberdinez egindako kopiak prest izan behar ditugula: egunekoak, astekoak, hilekoak, eta abar
- Hondamendi egoera (bai, dena galtzen den egoera hori) ere aurreikusi beharko genukeela kopiak erakundetik aterata.

**ElkarBackup** erakundearen datuen segurtasun kopiak kudeatzeko sistema izanik, administratzaileak bere eskura izango duen beste soluzio bat izango da, eta hauek dira bera ezaugarrietako batzuk:

- Web interfaze baten bidez kudeatzen da
- Kopiak diskora egingo ditu, ez zintetara, beraz azkarragoa izango da eta ez digu gailu fisiko **bereziekiko (zintak kudeatzeko gailuak)** dependentziarik sortuko.
- Esteka gogorrak (aurrerantzean HardLink<sup>1</sup>) erabiliko ditu, beraz fitxategiak diskoan okupatzen duten espazioa oso ondo optimizatuta egongo da.
- Interfazea euskara, gaztelania eta ingelesera lokalizatua dago.
- Interfazea autodokumentua izatea saiatu gara, ahal den neurrian eskuliburura jo beharrik ez izateko.

Software librean oinarritzen den tresna da, [GPL V3](#)<sup>2</sup> lizentziapean banatzen da, eta beste hainbat tresna libreetan oinarritzen da, batez ere rsnapshot<sup>3</sup> eta rsync<sup>4</sup>. GNU/Linux sistemetan instalatzen den arren (gaur egun Debian distribuziorako prestatua izan da), ssh/rsync exekutatu dezakeen edozein sistema eragiletako datuen kopiak egin ditzake.

Proiektu hau [IMH](#) eta [Tknika](#)ren elkarlanean garatu da 2012-2013 ikasturtean, ikastetxeei eta oro har beste edozein erakunderi kopiak modu zentralizatuan egiten laguntzeko, aurkitu izan dugun tresnak zituzten gabeziak gainditzeko asmoarekin. Garapena [Binovo](#) enpresak eginikoa da, eta testatzen [Elkarnet](#) taldeko partaideak aritu dira.

---

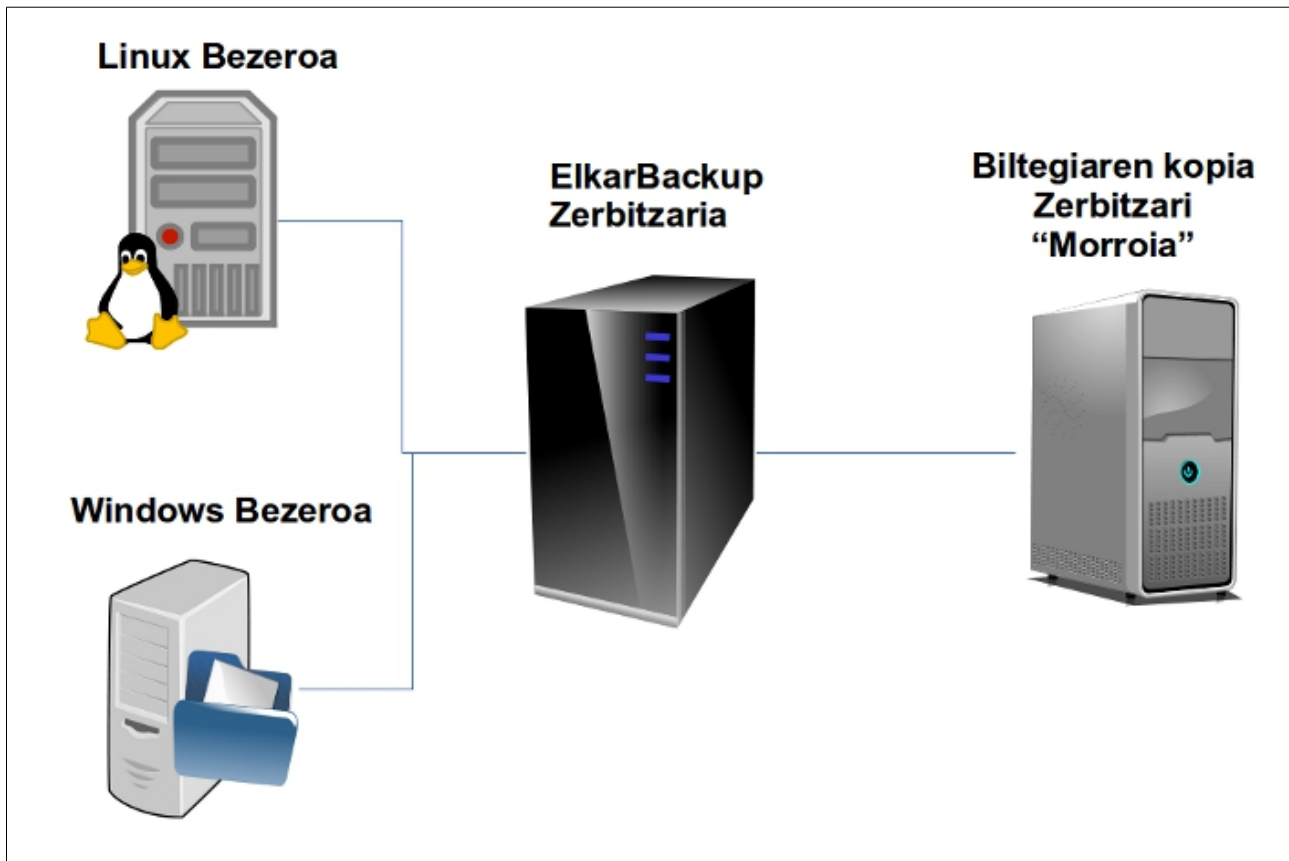
1 [http://es.wikipedia.org/wiki/Enlace\\_duro](http://es.wikipedia.org/wiki/Enlace_duro)

2 [http://en.wikipedia.org/wiki/GNU\\_General\\_Public\\_License#Version\\_3](http://en.wikipedia.org/wiki/GNU_General_Public_License#Version_3)

3 <http://www.rsnapshot.org/>

4 <http://es.wikipedia.org/wiki/Rsync>

Eskuliburu honetan azalduko dugun prozesuaren bukaeran ikusiko dugu nola instalatzen den ElkarBackup zerbitzaria, nola lortu GNU/Linux zerbitzari baten eta Windows Zerbitzari baten datuen kopiak egitea, eta azkenik datuen kopiaren emaitza den biltegia nola sinkronizatu daitekeen beste urruneko zerbitzari batean.



## 2.- Instalazioa

Tresna hau instalatzeko bi aukera daude, birtualizazio sisteman jartzeko prestatu dugun irudia deskargatu eta erabili, edo aurretik dugun Debian batean ElkarBackup paketea instalatzea.

### 2.1.- Guk prestatutako Irudia erabili

Debian batean oinarritutako instalazioa duen RAW irudia deskargatu eta zure birtualizazio sisteman gehitzea: Aukera desberdinak daude, guk proposatzen ditugunak Proxmox eta KVM dira, baina gehiago ere badaude: XEN, VMWare, eta abar. Dokumentu honen bukaeran dagoen ***"Irudia beste birtualizazio sistema batzuetara bihurtzen"*** atalean RAW irudia VMWare eta VirtualBox birtualizazio sistemetarako nola bihurtu daitekeen azaltzen dugu.

Azkeneko Irudiak hemendik deskargatu daitezke:

- 64 bitekoa: <ftp://ftp.binovo.es/tnika-backups/images/>
- 32 bitekoa:

Irudiak instalazioa egina duenez lanean hasteko martxan jartzea besterik ez dago. Hala ere, eta gure sarera egokitzeke pare bat gauza aldatu beharko ditugu.

Zerbitzariak DHCP bidez jasotzen du bere helbidea, eta hau aldatu beharra daukagu beti helbide berdina izatea interesatzen zaigulako. IP finkoa jartzeko ***/etc/network/interfaces*** fitxategia editatuko dugu

```
root@ElkarBackup:~# nano /etc/network/interfaces
```

Eta hor lpa, maskara, gateway eta DNSak jarriko ditugu.

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
# iface eth0 inet dhcp
iface eth0 inet static
address NIREIP
netmask MASKARA
gateway GATEWAY
```

```
dns-nameservers DNSak
```

Seguruenez arrankatzen duenean ez du sare txartela aktibatuko, bere txartela (kasu honetan birtuala) aldatu dela konturatzen delako. **70-persistent-net** rules fitxategia editatu eta bere edukia ezabatu. Gero berrabiarazi ondoren hau berak zuzenduko du sare txartel birtualaren informazio egokia jarritz.

```
root@ElkarBackup:~# nano /etc/udev/rules.d/70-persistent-net.rules
```

Makina birtualean **root** erabiltzaileak **root** pasahitza du. Hori ere produkzioko zerbitzarian aldatu beharko litzateke.

```
root@ElkarBackup:~# passwd root
Introduzca la nueva contraseña de UNIX:
Vuelva a escribir la nueva contraseña de UNIX:
passwd: contraseña actualizada correctamente
```

Aldaketa hauek egin ondoren makina birtuala berrabiarazi

```
root@ElkarBackup:~# shutdown -r now
```

## 2.2.- Aurrez instalatuta duzun Debian batean paketea instalatzea

Irudia erabili beharrean aurrez dugun Debian batean instalatu nahi badugu, kontutan izan hainbat pakete agian eskuz instalatu beharko direla. Elkarbackup paketeak besteak beste MySQL datu base bat behar du, beraz hemen aukeraketak egiteko unea heldu da:

- Edo aurrez gure ElkarBackup-Debian zerbitzarian **mysql-server** paketea instalatzen dugu berak datu basea ere kudeatu dezan
- Edo beste zerbitzari batean instalatuta daukagun MySQL zerbitzaria erabiltzen dugu datu basea kudeatzeko.

MySQL zerbitzarian backup sistemaren konfigurazio informazioa gordeko da: bezero eta lanen datuak, politikak, log-ak, erabiltzaileak, eta abar.

Hortaz aparte beste bakete batzuk derrigorrez instalatu behar dira: mysql bezeroa, PHP, eta abar. Guk hemen denak batera instalatuko ditugu (mysql zerbitzaria ere bai):

```
root@ElkarBackup:~# aptitude install debconf php5 php5-cli rsnapshot mysql-server php5-mysql acl bzip2
```

MySQL zerbitzaria instalatzerakoan bere **root** erabiltzailearen pasahitza eskatuko digu. Guk adibiderako **root** jarriko diogu, baina esan beharrik ez dago hau produkzioan aldatu beharko litzatekeela.

Orain ElkarBackup paketea instalatuko dugu. Paketeak dituen menpekotasunen artean, hauek nabarmenduko genituzke:

- Apache Web zerbitzaria: Erabiltzailearen web interfazea kudeatzeko erabiliko da
- Rsnapshot: HardLink-ak erabiliz diskora kopiak egiteko ardura izango duen softwarea
- ssh eta rsync: Bezeroekin konektatu eta datuak sinkronizatzeko erabiliko den softwarea.
- Fitxategi sisteman ACLak aktibatuta izan behar ditu.

Azken puntu honek garrantzia du: Sistemak ACLak instalatuta eta aktibatuta izan behar ditu gerora diskoan datuak gordetzeko erabiliko duen partizioan. Debian 7an hau horrela dator instalaziotik, hau da balio lehenetsi moduan ACLak aktibatut datoz, baina Debian 6an ez da horrela.

Elkarbackup paketea Debian 6an instalatzen ari bagara, **root partizioan (/)** ACL-ak aktibatu behar ditugu. Horretarako **/etc/fstab** fitxategia editatu eta **acl** jarriko dugu, ondoren azaltzen den moduan:

```
# / was on /dev/sda1 during installation
UUID=e3b77e85-df06-4659-b143-5939ccbf7d52 / ext3 errors=remount-ro,acl 0 1
```

Hau egin ondoren hoberena sistema berrabiraztea litzateke, hau ondo egin dugula ziurtatzeko.

ACL kontu hauek konpondu ondoren, lehenik eta behin errepositorioaren gakoa inportatu beharko dugu. Hau root erabiltzailearekin egin beharreko lana da:

```
root@backups:~# wget -O - ftp://ftp.binovo.es/tnika-backups/packages/repository_key.pub | apt-key add -
```

Hau egin ondoren **sources.list** editatu

```
root@backups:~# nano /etc/apt/sources.list
```

eta lerro hauek gehitu

```
# Binovo repository
deb ftp://ftp.binovo.es/tnika-backups/packages squeeze main
```

Azkenik eguneratu eta instalatu

```
root@ElkarBackup:~# aptitude update
root@ElkarBackup:~# aptitude safe-upgrade -y
root@ElkarBackup:~# aptitude install autofs elkarbackup
```

Instalazioan MySQL datu base bat sortu behar du aplikaziorako, beraz gure MySQLko admin erabiltzaile izena eta pasahitza eskatuko dizkigu. Gogoan izan horiek [MySQL zerbitzariaren instalazioan](#) emandako datuak direla, eta gure kasuan **root** erabiltzailea



**root** pasahitzarekin jarri dugula.

Azkenik beste ukitu bat. SSH konexioetan makina batera lehenengo aldiz konektatzen garenean, gure zerbitzariak esaten digu ez duela makina hori ezagutzen eta bere **fingerprint**<sup>5</sup> eskuz baieztatu behar dugula. Hau segurtasun neurri bat da, baina gure kasuan eta automatizazioari begira arazo bihurtzen da, beraz ElkarBackup zerbitzarian **/etc/ssh/ssh\_config** editatuko dugu eta hau gehituko diogu:

```
StrictHostKeyChecking no
```

Honela ez du galderarik egingo zerbitzari berri batera konektatzen denean.

Datu horiek sartu eta aplikazioaren instalazioa egingo da. Honekin paketea instalatuta eta lanerako prest izango dugu.

Kontutan izan ACLak aktibatuta ez baditugu paketearen instalazioak bukaeran errore hau emango digula:

```
ACLs not supported. Remount with ACL and reconfigure with 'dpkg --configure --pending'
dpkg: error al procesar elkarbackup (--configure):
  el subproceso instalado el script post-installation devolvió el código de salida de error 1
Se encontraron errores al procesar:
  elkarbackup
```

eta azalpenak dioen moduan, hau konpontzeko ACLak aktibatu eta ondoren instalazioa agindu honekin burutu daiteke:

```
root@elkarbackup:~# dpkg --configure --pending
```

## 2.3.- Eguneraketak

Instalazioa prozedura edozein dela ere, Debian zerbitzarian paketea eguneratzeko beste edozein pakete eguneratzeko erabiltzen den teknika berdina erabiliko da:

```
root@backups:~# apt-get update
root@backups:~# apt-get upgrade
```

<sup>5</sup> Informazio gehiago hemen: <http://linuxcommando.blogspot.com.es/2008/10/how-to-disable-ssh-host-key-checking.html>

---

### 3.- Zerbitzarira konektatzen

Orain web nabigatzaile bat zabaldu eta **ElkarBackup** zerbitzariaren helbidea jartzearekin nahiko litzateke aplikaziora konektatzeko. Hemen bi aukera ditugu:

- **Host** izena erabiltzea: Apache konfigurazioan **elkarbackup** izena **ServerName** atalean jarrita dugunez, <http://elkarbackup> helbidea erabili dezakegu atzipena lortzeko, beti ere gure sareko DNSan **elkarbackup** izena jarri dugun IP helbidearekin lotuta badaukagu.
- IP helbidea erabiltzea: Kasu honetan URLa honela osatu behar dugu: <http://IPHELBIDEA/elkarbackup/app.php/login>

Eta aplikazioaren **root** erabiltzailearekin sartzen gara, honek ere **root** pasahitza du (aplikazioak duen pasahitza aldatzeko aukeraren bidez aldatu dezakegu).

## 4.- Bezeroak eta Lanak

Ikusiko dugun lehen pantaila **Bezeroena** eta **lanena** da, eta noski, oraindik hutsik dago, baina laster bete beharko dugu.

Bezeroak	Politikak	Script-ak	Erabiltzaileak	Konfigurazioa	Logak	Saioa
----------	-----------	-----------	----------------	---------------	-------	-------

### Bezeroak

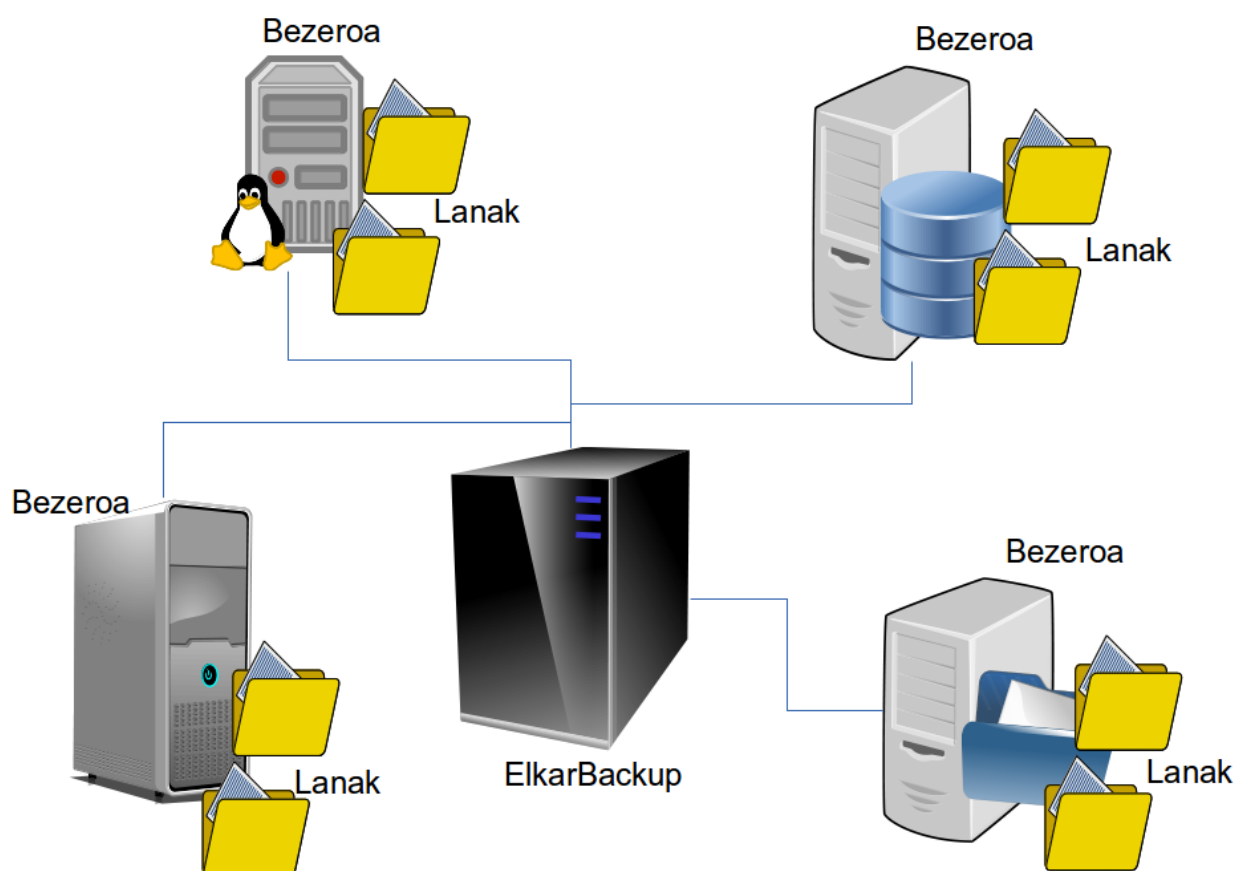
Id-a	Izena	Diskoaren erabilera	Azken log sarrera	Egoera	Ekintzak
<div>Berria</div>					

### 4.1.- Baina, zer dira bezeroak eta lanak ?

Arkitektura honetan **bezero** eta **lanei** buruz hitz egingo dugu:

- **Bezeroa**: GNU/Linux edo Windows makina bat. Makina honetan (gehienetan beste zerbitzari bat izango dena) dauzkagun karpeta ezberdinak kopiatu nahi ditugu.
- **Lana**: Bezero bakoitzetik kopiatu nahi dugun karpeta bakoitza (erroa). Bezero berdinekoak izan arren, lan bakoitzari politika desberdina esleitzeko aukera izango dugu, bakoitzari bere maiztasuna eta erretentzio politika definituz.

Adibidez, demagun zerbitzari baten alde batetik erabiltzaileen datuak ditugula, eta bestetik zerbitzu jakin batek sortzen dituen log fitxategiak. Denak kopiatu nahi ditugu, baina maiztasun eta erretentzio politikak zeharo desberdinak izango dira kasu batean eta bestean.



Ilustrazioa 1: Bezeraok eta lanak

Gure zerbitzariak hainbat bezeroetako informazioa jaso eta kopiatu ahal izango du. Bezeraok bete behar duen baldintza bakarra **rsync** edo **ssh** protokoloak erabiltzeko gaitasuna izatea izango da. GNU/Linux zerbitzariarekin ez da batere arazorik izango, eta Windows zerbitzarietan **Cwrsync**<sup>6</sup> softwarea instalatuko dugu rsync zerbitzurako (aurrerago azalduko dugu nola konfiguratu behar den).

Komunikaziorako rsync erabiltzeak onura garrantzitsu bat dauka: fitxategien transferentzia hasi aurretik, bien artean fitxategi bakoitza kopiatu beharra dagoen ala ez erabakitzen dute, eta aldaketak bakarrik bidaltzen dira. CPU lan gehien egin beharko duena fitxategiak bidaltzen dituen da eta S/I lan gehien egin beharko duena datuak jasotzen dituen da.

## 4.2.- GNU/Linux Bezeraok gehitzen

Bezera bat gehitzeko, **Berria** botoiaren gainean klikatu eta formulario berri bat azalduko zaigu. Aplikazioa autodokumentatua izatea saiatu gara, eta azalpen gehienak formularioan bertan daudenez, ez ditugu hemen azalpenak errepikatuko.

6 <https://www.itefix.no/i2/content/cwrsync-free-edition>

Gure adibidean erabiliko ditugun GNU/Linux zerbitzariaren datuak hauek izango dira:

- Izena: Debian Bezeraok
- URL: root@10.15.181.155
- KUOTA: -1 (ez dugu oraingoz kuotarik ezarriko)
- Azalpena: Gure sareko Linux zerbitzari bat
- Pre/Post script: ezer aukeratu gabe

Menuan **Bezeraok → Erakutsi** klikatzen badugu ikusiko dugu bezeraok zerrendan agertzen dela, nahiz eta oraindik lanik ez izan.

Bezeraok	Politikak	Script-ak	Erabiltzaileak	Konfigurazioa	Logak	Saioa
<b>Bezeraok</b>						
Id-a	Izena	Diskoaren erabilera	Azken log sarrera	Egoera	Ekintzak	
1	Debian Bezeraok	0 MB		Aktibo	Ezabatu	
						Berria

Baina, ...., kontutan izanik backupak egiteko komunikazioak automatizatuak izatea nahi dugula, **ElkarBackup zerbitzua nola konektatuko da bezeraok honekin inon ere bere pasahitzik jarri ez badugu ?**

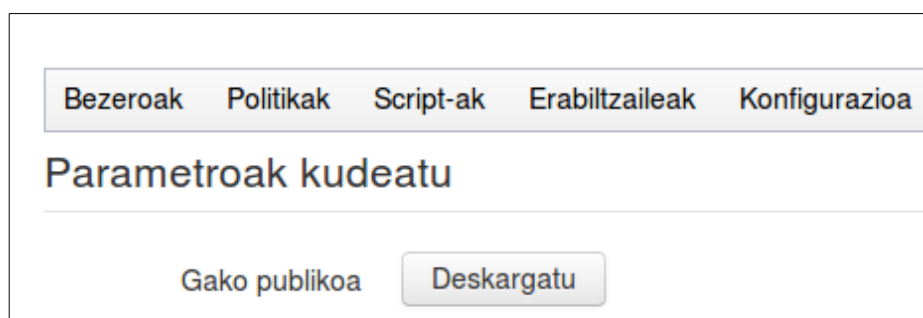
Komunikazioa ssh protokoloaren bidez izango direnez, gako publiko/pribatuak erabiliko ditugu.

Menu nagusian: **Konfigurazioa → Parametroak kudeatu → Generate** botoiaren gainean behin behintzat klikatu behar dugu zerbitzariak bere gakoa sortu dezan

Bezeraok	Politikak	Script-ak	Erabiltzaileak	Konfigurazioa	Logak
<b>Parametroak kudeatu</b>					
Gako publikoa		Generate		MySQL kon	

Aurrerantzean gako publikoa deskargatzeko aukera izango dugu.

Menu nagusian: **Konfigurazioa** → **Parametroak kudeatu** → **Gako publikoa deskargatu**



**Deskargatu** botoiaren gainean klikatu eta **Publickey.pub** fitxategian Elkarbackup zerbitzariaren gako publikoa deskargatuko dugu. Gako hau gure GNU/Linux bezeroetan instalatu beharko dugu, eta horrela ElkarBackup zerbitzariak datuak kopiatzerako orduan konexio automatizatuak abiatu ahal izango ditu.

Segurtasun kopian SSH bidez datuak kopiatzen direnean rolak honela banatzen dira:

- ElkarBackup zerbitzaria: konexioa zabalduko duen **bezeroa** izango da.
- GNU/Linux bezeroa: ssh konexioaren **zerbitzaria** izango da, beraz **openssh-server** paketea instalatuta izan beharko du (kasu gehienetan instalatuta izango dugu).

Gure adibidean honela instalatuko dugu **Publickey.pub** gakoa bezeroan. Erabiltzaile moduan web nabigatzailea erabiliz nire ekipora deskargatzen dut, eta **ssh-copy-id** komandoa erabiliz bezeroaren **root** erabiltzailean instalatzen dut. Normala den bezala hau egin ahal izateko bezeroaren root erabiltzaileak duen pasahitza ezagutu behar dut.

```
pedro@portatil59:~/Deskargak$ ssh-copy-id -i Publickey.pub root@10.15.181.155
root@10.15.181.155's password:
Now try logging into the machine, with "ssh 'root@10.15.181.155'", and check in:

~/ .ssh/authorized_keys

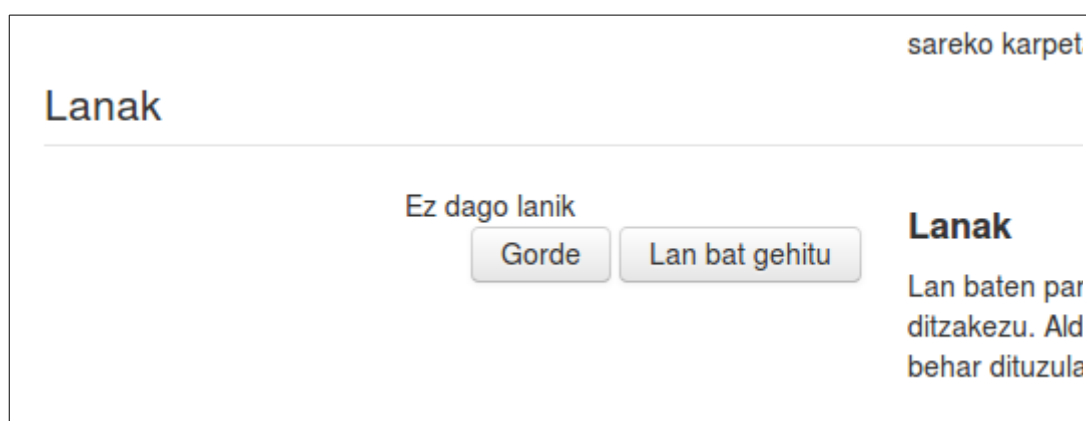
to make sure we haven't added extra keys that you weren't expecting.
```

Egin dugunarekin bezeroaren **root** erabiltzaileko **ssh** konfigurazioan ElkarBackup zerbitzariak sortutako gakoa gehitu dugu, eta aurrerantzean bien arteko konexio automatizatuak abiatu ahal izango ditugu.

```
root@DebianBezeroa:~# ls -la /root/.ssh/authorized_keys
-rw----- 1 root root 428 jun  7 11:11 /root/.ssh/authorized_keys
root@DebianBezeroa:~# cat /root/.ssh/authorized_keys
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQAC3H03zn8XhBW1JqxASqKEQZe+/fPcC47pu5l9c+s1Q+ppUD5C
```

LaDQjUsF+0VBHtPP2Wx1HaDidFURwge5GcnRnoXV32B+Vj9rfW9wPdk6siuZ2McoS0xyqbToS2CTdbS  
 xyjibn2CdM3RZjJa81ha0llciu38V9F1t4mylJVMxBgQmNwkTBwYGt66+wpT/nQVxiDeyVK98SHb8oG  
 KJZljpczlqgYwckRAEPSARvxFm9yyW0ad3Qm7QYYGslBij4LXGlaUAARivoJdYtB4kb0nEd003n5BU/  
 0Q8eIqx Dw7eFdnx4UDINX+mtSuHfpRko0RoU9IZZWGz9vrLnaxqh91G3 Automatically  
 generated key for tknikabackups.

Orain lan berri bat sortuko dugu Debian bezeroan daukagun **/media/Backups** karpeta kopiatzeko. Horretarako aurrez sartu dugun bezeroaren konfigurazioa editatu eta beheko aldean dagoen **Lan bat gehitu** botoiaren gainean klikatuko dugu



Agertzen zaigun formularioa ere autodokumentatua dago eta nahiko ondo ulertzen da sartu beharreko informazioa zein den. Guk datu hauek sartuko ditugu, beste guztia dagoen moduan utzita:

- Izena: Backups karpeta
- Path: /media/Backups/
- Azalpena: Backups karpetan ditugun datuak
- Politika: Default policy

Eta **Gorde** botoiari emango diogu konfigurazioa gorde dezan. Orain berriro Bezeroen ikuspegi orokorrera joaten bagara lana hor dagoela ikusiko dugu.

Bezeroak
Politikak
Script-ak
Erabiltzaileak
Konfigurazioa
Logak
Saioa

## Bezeroak

Id-a	Izena	Diskoaren erabilera	Azken log sarrera	Egoera	Ekintzak
1	Debian Bezeroa	0 MB		Aktibo	Ezabatu
1.1	Debian Bezeroa/Backups karpeta	0 MB		Aktibo	Ezabatu Restore

Berria

Inoiz exekutatu ez denez **Azken log sarrera** zutabea utzik dago. Aurrerantzean zutabe horretan lan honen azken exekuzioko emaitza zein eta noiz izan den ikusi ahal izango dugu.

Baina noiz izango ditugu datuak ? Ba hori **Politika** planifikazioan jarritakoaren arabera izango da. Guk **Default policy** politika jarri diogu, eta hor zer dagoen ikustera joaten bagara, ikusiko dugu zein den definituta daukan programazioa eta erretentzio politika.

Politiken nondik norakoa ondo ulertzeko **“Oinarrien errepaso txiki bat: Rsnapshot”** atalean azaldutako kontzeptuak argi izatea komeni da, azken finean hori delako hemen aplikatuko den logika. Bukaeran sakonduko dugu gai horretan.

Lehen kopia egiteko tarteak utzi eta **Restore** botoian klikatuz zer gertatu den ikustera joaten bagara



Bezeroak
Politikak
Script-ak
Erabiltzaileak
Konfigurazioa
Logak
Saioa

## Debian Bezeroa/Backups karpeta//

[Bezerora bueltatu](#)  
[Lanera bueltatu](#)

/var/spool/elkarbackup/backups/0001/0001

Izena	Aldaketa data	Ekintzak
.	2013-06-10 09:24:02	
..	2013-06-07 12:00:02	
.sync	2013-06-10 09:24:02	<a href="#">.tgz bezala deskargatu</a>
Hourly.0	2013-06-10 09:24:02	<a href="#">.tgz bezala deskargatu</a>
Hourly.1	2013-06-10 09:20:03	<a href="#">.tgz bezala deskargatu</a>
Hourly.2	2013-06-10 09:00:10	<a href="#">.tgz bezala deskargatu</a>

ikus dezakegu Hourly.0 karpeta hor dagoela azken kopiaren datuekin, eta hurrengoko exekuzioekin karpeta gehiago pilatzen joango dira. Karpeta barruan murgiltzen bagara, guk nahi dugun fitxategi zehatz batera heldu gaitzke

Bezeroak
Politikak
Script-ak
Erabiltzaileak
Konfigurazioa
Logak
Saioa

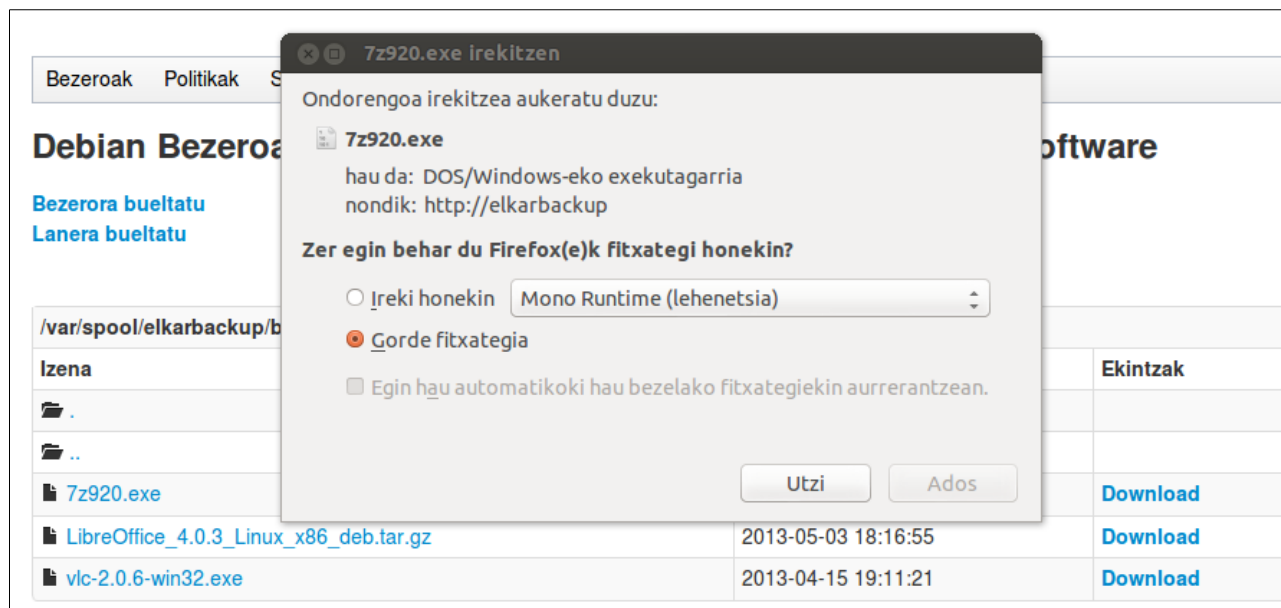
## Debian Bezeroa/Backups karpeta///Hourly.0/media/Backups/Software

[Bezerora bueltatu](#)  
[Lanera bueltatu](#)

/var/spool/elkarbackup/backups/0001/0001/Hourly.0/media/Backups/Software

Izena	Aldaketa data	Ekintzak
.	2013-06-07 11:24:19	
..	2013-06-07 11:24:47	
7z920.exe	2010-11-18 20:01:02	<a href="#">Download</a>
LibreOffice_4.0.3_Linux_x86_deb.tar.gz	2013-05-03 18:16:55	<a href="#">Download</a>
vlc-2.0.6-win32.exe	2013-04-15 19:11:21	<a href="#">Download</a>

Fitxategi horren kopia deskargatzeko, estekan klikatzea besterik ez da behar:



### 4.3.- Windows Bezeroak gehitzen

Windows eta Linux bezeroen arteko desberdintasuna **URL** parametroan dator, beste guztia berdina da.

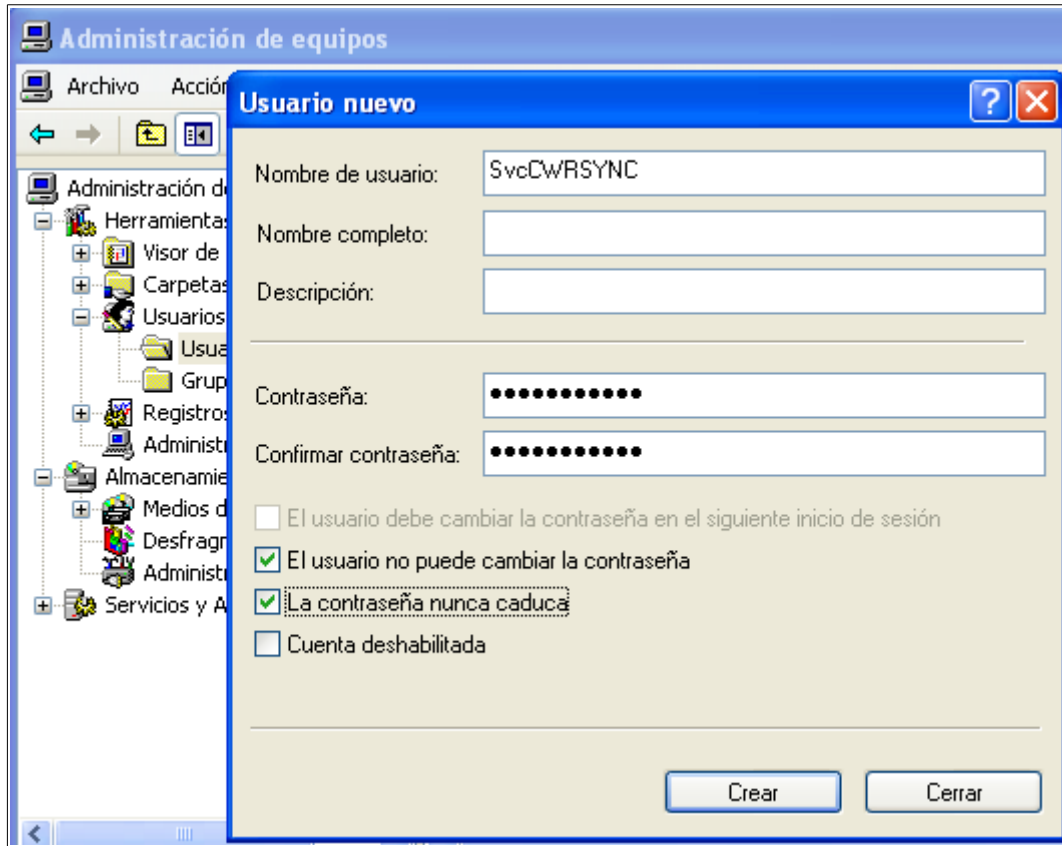
Windows zerbitzariekin ez dugu **ssh** protokoloa erabiliko, **rsync** baizik. Hau egin ahal izateko Windows makinak rsync konexioak onartzeko prestatuta egon behar da, hau da, rsync zerbitzari bat martxan izan behar du.

- Izena: Windows Bezeroa
- URL: **10.15.181.156:**
- KUOTA: -1
- Azalpena: Gure sareko Windows zerbitzari bat
- Pre/Post script: ezer aukeratu gabe

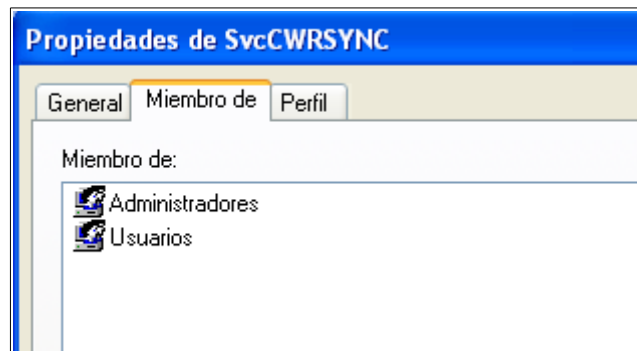
URL parametroan IPa jarri ondoren “:” jarri ditugu. Horrela adierazten diogu rsync protokoloa erabili behar duela.

Rsync zerbitzari desberdinak aurkitu ditzakegu Windows plataforman instalatzeko. Guk doako bertsioa duen **cwRsync** erabiliko dugu. Horretarako **cwRsyncServer 4.0.5 Installer** izeneko programa deskargatuko dugu.

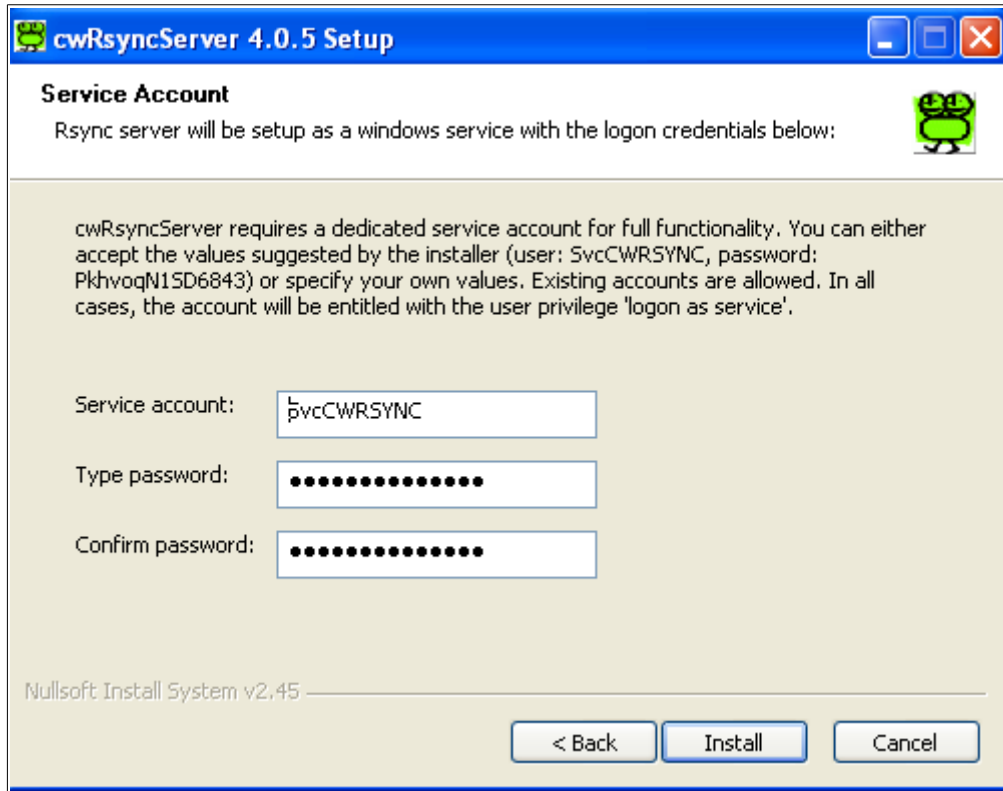
Instalazioa egin aurretik, zerbitzarian erabiltzaile lokal bat gehituko dugu. Erabiliko dugun WindowsXP makina birtualean nik **SvcCWRSYNC** deitu diot, eta **elkarbackup** pasahitza jarri diot.



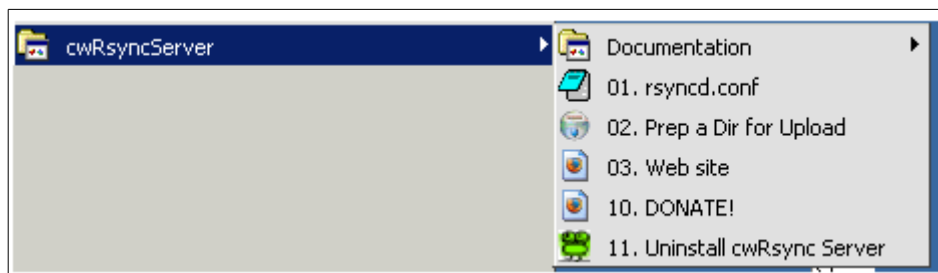
Zerbitzua erabiltzaile honekin exekutatu da, beraz baimenak izan beharko dituen Administrazioaren taldean sartu dut (agian baimen gutxiagokin ere nahikoa izan daiteke).



Orain deskargatu dugun **cwRsyncServer** instalatu dugu. Zerbitzuentzako erabiltzaile eta pasahitza eskatu dizkigu, eta oraintxe bertan sortu ditugunak erabiltzeko esango diogu:



Instalazioa bukatzen denean programen artean beste bat aurkituko dugu, eta hor ikusten dugun **rsyncd.conf** fitxategia editatuko dugu sinkronizatu nahi ditugun karpetak zeintzuk diren zehazteko



Ondorengo irudian ikusten da nola utzi dudun [C:\Backups](#) karpeta konfiguratzeko.

Hasieran bi lerro jarriko ditugu:

```
uid=0
gid=0
```

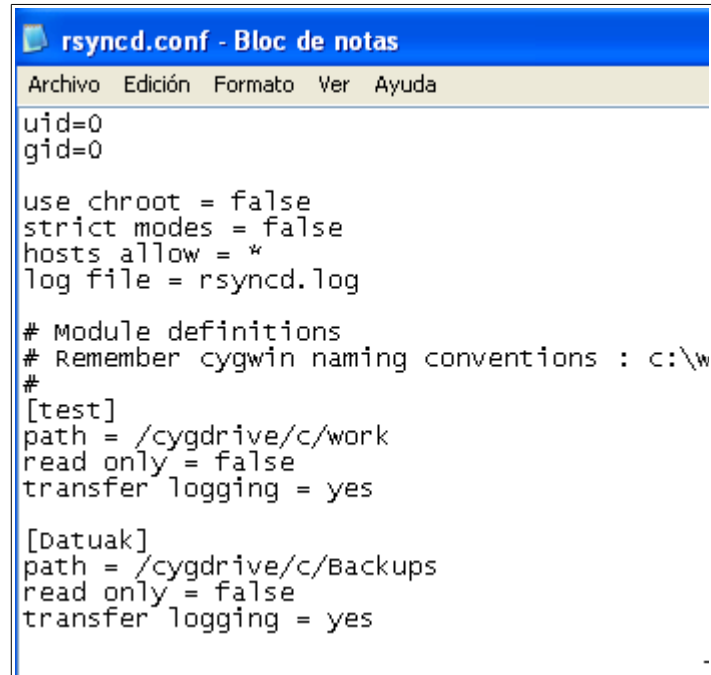
Ondoren sinkronizatu nahi dugun karpeta bakoitzeko bloke bat gehitu behar dugu. Adibidean blokeari **[Datuak]** izena jartzen diogu:

```
[Datuak]
path = /cygdrive/c/Backups
```

```
read only = false
transfer logging = yes
```

Kopiatu nahi dugun karpeta [c:\Backups](#) izan beharrean [d:\Backups](#) izan balitz, path lerroa horrela geratuko litzateke

```
path = /cygdrive/d/Backups
```






```
rsyncd.conf - Bloc de notas
Archivo Edición Formato Ver Ayuda
uid=0
gid=0

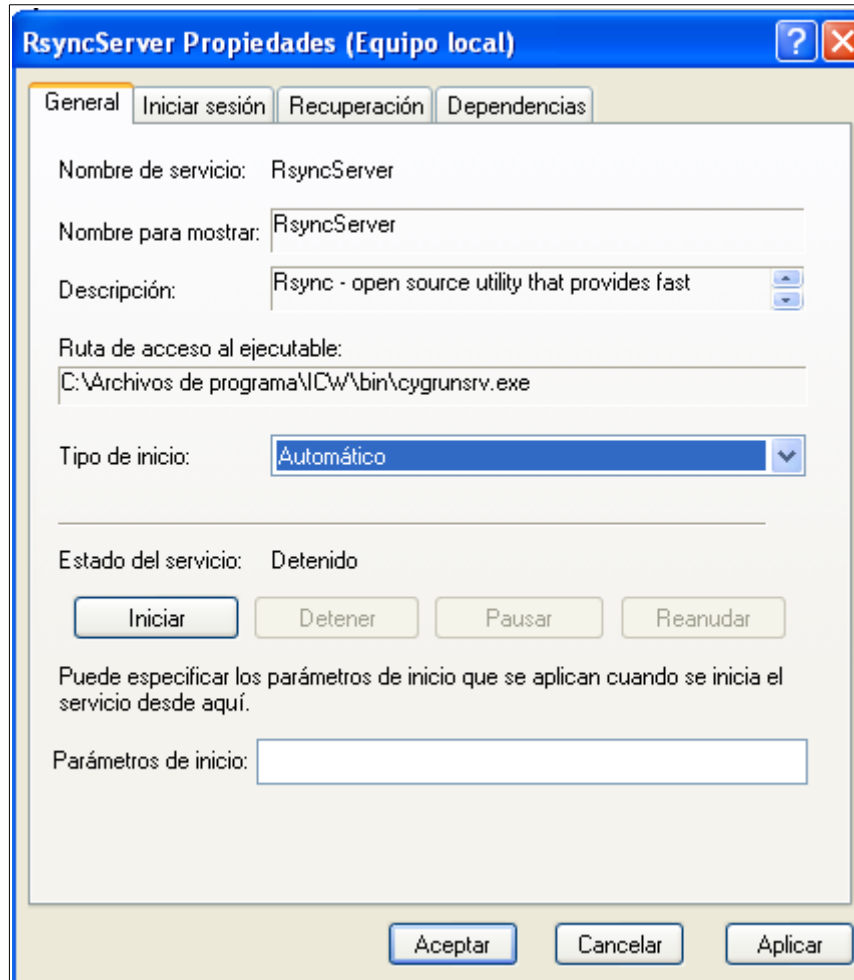
use chroot = false
strict modes = false
hosts allow = *
log file = rsyncd.log

# Module definitions
# Remember cygwin naming conventions : c:\w
#
[test]
path = /cygdrive/c/work
read only = false
transfer logging = yes

[datuak]
path = /cygdrive/c/Backups
read only = false
transfer logging = yes
```

Orain zerbitzua martxan dagoela ziurtatu, eta ez badago eskuz abiatu eta automatiko jarri. Horretarako zerbitzuetan begiratu

	Registros y alertas de rendimie...	Recopila información de ren...	Manual	Servicio de red
	RsyncServer	Rsync - open source utility t...	Iniciado	Automático
	Servicio COM de grabación de ...	Administra la grabación de ...	Manual	Sistema local



Orain ElkarBackup interfazera bueltatzen gara eta gehitu berri dugun Windows bezeroari lan bat gehituko diogu, Backups karpeta kopiatzeko lana hain zuzen ere.

Gogoan izan Windows makinako konfigurazio fitxategian [**Datuak**] izeneko bloke bat sortu dugula, hori izango da orain **Path** eremuan erabiliko dugun izena, eta **Default policy** politika aplikatuko diogu

### Lana editatu

**Backupak Bezeroa**

Izena

Path

Diskoaren erabilera

Azalpena

Aktibo dago ☒

Gordeko dugu eta ondoren **Exekutatu orain** botoiaren gainean klikatuko dugu, ea dena ondo egiten duen probatzeko. Botoi honek kopia abatzeko aukera emango digu bere programazioaren zai egon gabe.

Kontutan izan lanen ikuspegi orokorra ematen digun pantailak askotan lanaren egoera ikusteko balio izango digula, adibidez zain dagoenean (**QUEUED**) edo exekuzioan dagoenean (**RUNNING**)

Bezeroak					
Id-a	Izena	Diskoaren erabilera	Azken log sarrera	Egoera	Ekintzak
1	Debian Bezeroa	187 MB	2013-06-07 16:41:03 OK	Aktibo	<button>Ezabatu</button>
1.1	Debian Bezeroa/Backups karpeta	187 MB	2013-06-07 16:41:03 OK	Aktibo	<button>Ezabatu</button> <button>Restore</button>
2	Windows Bezeroa	0 MB	2013-06-07 19:18:02 RUNNING	Aktibo	<button>Ezabatu</button>
2.2	Windows Bezeroa/Backups karpeta	0 MB	2013-06-07 19:18:02 RUNNING	Aktibo	<button>Ezabatu</button> <button>Restore</button>

Berria

Minutu bat itxaron eta kopia eginda aurkituko dugu. Nahi izan ezkerro, Log-etara joan eta prozesuaren emaitza ikusterik ere badugu

INFO	TickCommand		Command success: {"command": "elkarbackup:run_job", "client": "2", "job": "2"}	
INFO	StatusReport		OK	/client/2
INFO	RunJobCommand		Client "2", Job "2" du end.	/client/2/job/2
INFO	RunJobCommand		Client "2", Job "2" du begin.	/client/2/job/2
INFO	StatusReport		OK	/client/2/job/2
INFO	RunJobCommand		Client "2", Job "2" ok.	/client/2/job/2
INFO	RunJobCommand		Command "/usr/bin/rsnapshot" -c "/tmp/rsnapshot.2_2.cfg" Hourly 2>&1 succeeded with output:	/client/2/job/2
INFO	RunJobCommand		Running "/usr/bin/rsnapshot" -c "/tmp/rsnapshot.2_2.cfg" Hourly 2>&1	/client/2/job/2
INFO	RunJobCommand		Command "/usr/bin/rsnapshot" -c "/tmp/rsnapshot.2_2.cfg" sync 2>&1 succeeded with output:	/client/2/job/2
INFO	RunJobCommand		Running "/usr/bin/rsnapshot" -c "/tmp/rsnapshot.2_2.cfg" sync 2>&1	/client/2/job/2
INFO	StatusReport		RUNNING	/client/2/job/2

#### 4.4.- Akatsak konpontzen

Kontutan izan behar da zerbitzari ezberdinen arteko komunikazioa erabiltzen dugunez, arazoak suertatu daitezkeela komunikazio hauetan. Arazo hauen jarraipena **Logak** atalaren bitartez egin ahal izango dugu.

Dokumentu hau idazterakoan arazo bat izan dut Windows bezeroaren komunikazioarekin, eta mezu hau ikusten nuen **Logak** atalean:

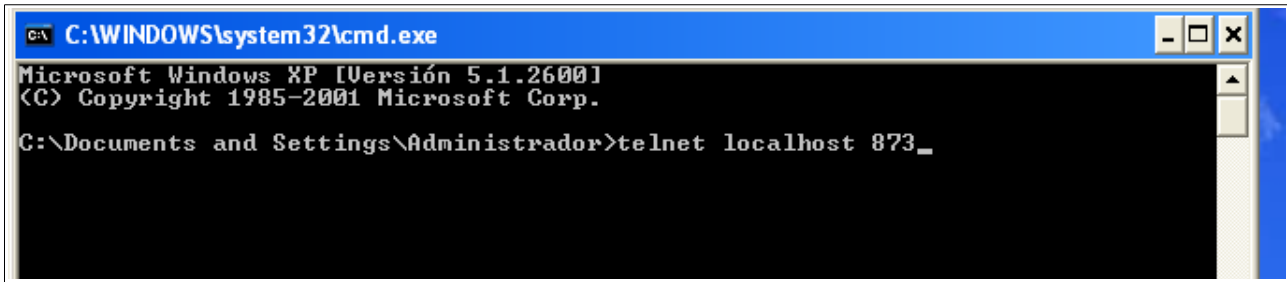
```
Command "/usr/bin/rsnapshot" -c "/tmp/rsnapshot.2_2.cfg" sync 2>&1 failed.
Diagnostic information follows: rsync: failed to connect to 10.15.181.156:
Connection timed out (110) rsync error: error in socket IO (code 10) at
clientserver.c(122) [Receiver=3.0.7]
```

Sistemak esaten dit arazoak izan dituela bezeroarekin rsync bidez komunikatzeko. Honek gutxienez bi arrazoi izan ditzake:

1. Windows bezeroan rsync zerbitzua ez dabil ondo
2. Bezera horrekin Rsync protokoloarekin komunikazio arazoak dauzkagu

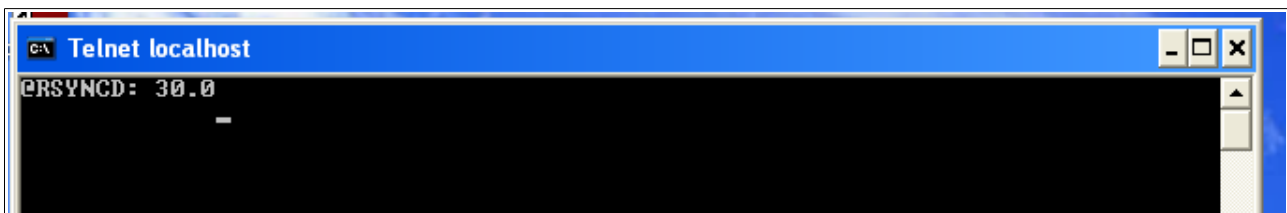
Lehenengo arazoa ote den jakiteko, rsync portura (TCP 873) telnet bat egiten saiatu naiz windows bezeroan bertan.





```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Versión 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\Documents and Settings\Administrador>telnet localhost 873_
```

eta bere erantzuna hau bada, zerbitzua ondo dagoela esan nahi du



```
Telnet localhost
ERSYNCD: 30.0
-
```

Ordoren konexio saiakera berdina egin dut baina oraingoan ElkarBackup zerbitzaritik Windows bezerora:

```
root@ElkarBackup:~# telnet 192.168.3.109 873
Trying 192.168.3.109...
telnet: Unable to connect to remote host: Connection timed out
```

ikusten dugunez ezin da komunikazioa ezarri, eta itxura osoa du arazoa suebaki baten egongo dela, ziurrenez Windows bezeroaren suebakian.

Windows bezerora joan eta bere suebakia desgaitu ondoren arazoa konpondu da. Honek ez du esan nahi bezeroen suebakiak desgaitu behar direnik, baina gutxienez **rsync** protokoloa (TCP 873) gaituta izan beharko genuke ElkarBackup eta Windows bezeroen artean.

## 5.- Politikak

Orain arte egin dugunarekin lortu dugu kopiak egitea, baina segurtasun kopia batean gauza gehiago ere zehaztu behar dira:

- Programazioa: Noiz egingo dira kopiak ?
- Erretentzioa: Kopia zaharrak noiz arte mantendu behar dira ?
- Maiztasun ezberdinak nahastu behar dira ? Egunean zehar, egunero, astero, edo hileroko kopiak izan behar ditugu ?

Gauza guzti hauek zehazteko **politikak** erabiliko ditugu (orain arteko adibideetan **Default Policy** izenekoa erabili dugu). ElkarBackup aplikazioa rsnapshot softwarean oinarritzen denez merezi du **"Oinarrien errepaso txiki bat: Rsnapshot"** atalean azaltzen dena errepasatzea kontzeptuak argi izateko. Laster egingo dugu.

Aplikazioak politika ezberdinak definitzeko aukera ematen digu. Lan bakoitzari politika bat esleituko diogu, eta politika bakoitza lan ezberdinetan berrerabiltzeko aukera izango digu.

Hauek dira politika berri bat gehitzerakoan eman beharko ditugun datuak:

- Izena eta Azalpena: testu libreko eremuak
- Baztertu: Kopiatik baztertuko ditugun fitxategien patroiak.
- Sartu: Kopian sartuko ditugun fitxategien patroiak, nahiz eta **Baztertu** eremuan jarritako patroiarekin bat egin. Adibidez, demagun orokorrean ez ditugula bideo fitxategiak kopian sartu nahi, eta horretarako \*.avi jarri dugula **Baztertu** eremuan, baina zuzendariaren karpeta dauden avi fitxategiak derrigorrez kopiatu behar direla. Kasu honetan **Sartu** eremuan **zuzendaria/\*.avi** jarriko genuke salbuespena gauzatzeko.
- Lehenengo sinkronizatu: rsnapshot programak kopia berri bat egiten duenean ordurarte dauden karpeten errotazioa egin eta karpeta baten (adibidez Daily.0) gauzatzen du azken kopia, **aldatuak izan diren** nahiz **aldatuak izan ez diren** fitxategiak karpeta bakar batean uzten dituelarik. Aldatu ez diren fitxategiak **hardlink** bitartez estekatzen ditu, eta horrela diskoan ez dute erabilitako espazioa bikoizten.

Prozesu honek denbora eskatzen dio, karpeta berrian fitxategi bakoitza kokatzerakoan kopia berri bat den edo hardlink bidez estekatu behar duen erabaki behar duelako, eta prozesu guztia hasi eta amaitu bitartean, urruneko bezeroa berari atenditzen dago. Kasu batzuetan ez du garrantzirik izango, baina beste batzuetan garrantzitsua izango da bezeroa ahalik eta azkarren libre uztea. Kasu hauetan **Lehenengo sinkronizatu** aukeratu dezakegu.

Hau aukeratzen dugunean, rsnapshot-ek egingo duen lehenengoko gauza zera izango da, **.sync** izeneko karpeta batean bezeroaren karpeta osoa sinkronizatzea, bezeroa ahalik eta azkarren libre uzteko. Prozesu hau bukatzen denean, eta bezeroa libre utzi ondoren, azkeneko kopian (adibidez Daily.0) zeuden fitxategiak **.sync** karpetan dauzkan fitxategiekin alderatzen hasiko da, eta estruktura berri osoa eraikiko du, behar diren hardlink-ak jarritz.

Beraz, eta izenak dioen logikari erantzunez, hau aukeratzen dugunean beste ezer egin aurretik sinkronizazio oso bat egingo du, eta ondoren kapetak osatzen eta behar diren errotazioak egiten hasiko da.

### Programazioa

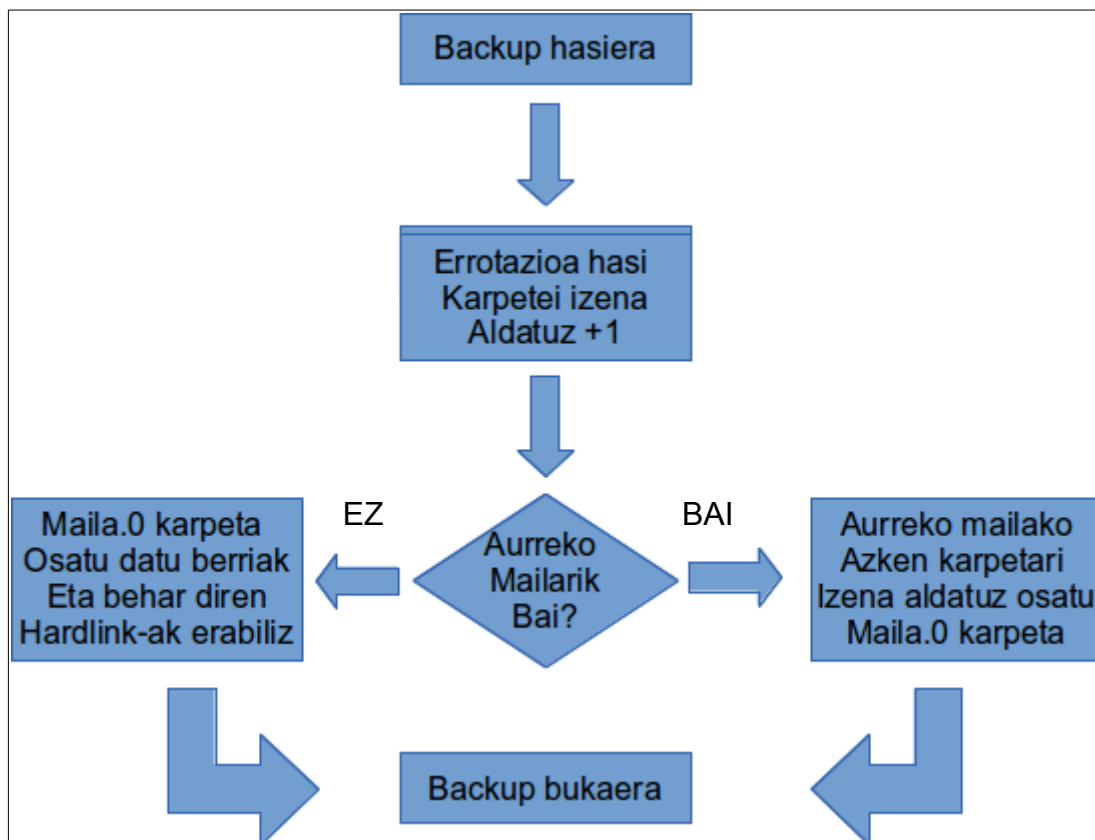
Hau da ulertu beharreko kontu garrantzitsuenetako bat. Alde batetik programazioa konfiguratzeko area ikus dezakegu, 5 fitxaz osatzen dena:

- Hourly edo Egunean zehar: Egunaren barruan, ordu ezberdinetan egingo dituen kopiak.
  1. Lehenengo Hourly karpeten errotazioa egingo du, azkena ezabatu eta besteei beti +1 eginez
  2. Hourly.0 karpeta sortu eta osatu: fitxategi berriak eta behar diren hardlink-ak
- Daily edo Egunero: Egunean behin ordu zehatz batean, eta aukeratutako asteko egunetan egingo duen kopia. Kontutan izan politika honen barruan aurreko mailan kopiak aktibatuta daudenean, hemen ez direla kopia berririk egingo, errotazioa baizik: Azken Hourly kapeta Daily.0 bihurtuko delarik.
  1. Lehenengo Daily karpeten errotazioa egingo du, azkena ezabatu eta besteei beti +1 eginez
  2. Aurreko maila (Hourly) erabiltzen da ?
    - Bai: Daily.0 osatzeko azken Hourly karpetari izena aldatu
    - Ez: Daily.0 karpeta sortu eta osatu: fitxategi berriak eta behar diren hardlink-ak.
- Weekly edo astero: Astean behin astegun eta ordu zehatz batean egingo duen kopia. Kontutan izan politika honen barruan aurreko mailan kopiak aktibatuta daudenean, hemen ez direla kopia berririk egingo, errotazioa baizik: Azken Daily kapeta Weekly.0 bihurtuko delarik.
  1. Lehenengo Weekly karpeten errotazioa egingo du, azkena ezabatu

eta besteei beti +1 eginez

## 2. Aurreko maila (Daily) erabiltzen da ?

- Bai: Weekly.0 osatzeko azken Daily karpetari izena aldatu
- Ez: Weekly.0 karpeta sortu eta osatu: fitxategi berriak eta behar diren hardlink-ak.
- Monthly edo Hilero: Goian aipatutako logika berdina errepikatzen da
- Yearly edo urtero: Goian aipatutako logika berdina errepikatzen da



Maila bakoitzeko **azken karpeta** aipatu dugu behin baino gehiagotan. Honek zer ikusia du erretentzio politikarekin. Esaten dugunean maila baten erretentzioa N dela, horrek esan nahi du maila horretan sortuko diren karpetak 0tik (N-1)erarte izendatuko direla.

Adibidez, egunean zehar egin beharreko kopiak 4ko erretentzioa izango dutela esaten badugu, sistemak 4 karpeta gordeko ditu: Hourly.0, Hourly.1, Hourly.2 eta Hourly.3 gordeko ditu.

Hourly errotazioa heltzen denean hau egingo du:

Azken Hourly-a ezabatu:

```
rm -Rf Hourly.4
```

beste guztiei izena aldatu errotazioa osatuz

```
mv Hourly.3 Hourly.4
mv Hourly.2 Hourly.3
mv Hourly.1 Hourly.2
mv Hourly.0 Hourly.1
```

eta azkenik Hourly.0 berri bat sortu eta bere edukia osatu, horretarako fitxategi berriak berataraz kopiatu eta behar diren Hardlink-ak sortuz.

### Arrasto zaharrak

Demagun Politika baten hourly-ak 6ko erretentzioa duela, horrek esan nahi du Hourly.0, ..., Hourly.5 karpetak izango ditugula kopia ezberdinekin. Arrazoiren batengatik, erabakitzen dugu gehiegizkoa dela eta 4ko erretentzioa izan behar duela, beraz Hourly.0, ..., Hourly.3 karpetak erabiliko dituela.

Zer gertatzen da Hourly.4 eta Hourly.5 karpetekin ? Ba inork ezabatzen ez baditu hor egongo dira. Ez dira eguneratuko, zaharkituta geratuko dira eta ez dute errotazioetan parte hartuko, baina hor egongo dira **"zarata egiten"**. Hoberena guk geuk eskuz sistematik ezabatzea litzateke.

Eta ez litzateke hobeagoa kasu hauetan sistemak karpeta hauek automatikoki ezabatzea? Agian bai, baina datuak automatikoki ezabatzeak ere arriskuak ditu. Demagun **produkzioan** dagoen politika baten aldaketak egiten ari garela eta konturatu gabe parametro horretan aldaketa bat egiten dugula erretentzio txikiago bat jarritz, eta ondorioz automatikoki karpetak ezabatzen dituela. Datuen galera ahal den neurrian ekidin beharko lirateke, beraz hobe eskuz ezabatzea.

### Politika ezberdinen erabilera

Gure sareetan informazio mota ezberdina dugu, eta guztia kopiatu beharra izateak ez du esan nahi denak trataera berdina izan behar duenik.

Kasu batzuetan informazioa epe luzeetan mantentzekoa izango da (hileak edo urteak), beste kasu batzuetan aste bete igaro ondoren ez du baliorik izango eta ezabatu nahiko genuke.

Hortaz gain, datu mota batentzat nahikoa izango da gaueko kopia izatea, eta agian beste

---

datu mota batentzat egunean 3 kopia izatea nahiko genuke.

Behar desberdin hauetarako politika ezberdinak sortu ditzakegu, eta nahikoa izango da **lan** bakoitzari dagokion politika esleitzea.

Kontutan izan **Bezero** batean lan ezberdinak politika ezberdinekin programatuta egon daitezkeela, **politika lanarekin lotzen baita**, ez bezeroarekin.

## 6.- Scriptak

Kasu askotan lan baten exekuzioa hasi aurretik edo bukatu ondoren **zerbait** egitea interesatuko zaigu, adibidez

Aurretik:

- Agian VPN konexio bat zabaldu behar dugu urruneko zerbitzari batekin bere datuetara atzipena lortzeko.
- Bezeroan sistema osoaren [snapshot](#)<sup>7</sup> bat sortu datuak interferentziarik gabe lasai kopiatzeko
- Zerbitzu bat geratu bere datuak kopiatu ahal izateko (zimbra, MySQL, ....)
- eta abar.

Ondoren:

- Zabaldu dugun VPN hori berriro ere itxi
- Hainbat datu koherenteak direla ziurtatu: data egokiak, bzip fitxategiak ondo deskonprimitzen direla, ....
- eta abar

Ekintza hauek programatu ahal izateko scriptak erabiliko ditugu, eta guztiz autonomoak izango gara guk behar ditugun scriptak garatu, eta nahi badugu besteekin elkarbanatzeko.

### 6.1.- Script berri bat

Hauek dira script bat sortzerakoan eskatzen dizkigun datuak:

- Izena eta azalpena
- Fitxategia: gure ordenagailuan programatu dugun scripta igotzeko aukera
- Exekuzio aukerak: Script bakoitzaren logikaren arabera, batzuk **Bezero mailan** pentsatuak izango dira, eta beste batzuk **Lan mailan**. Hauetako batzuk **Pre-script** moduan (ekintza abiatu aurretik exekutatu) eta beste batzuk **Post-script** moduan (ekintza bukatu ondoren exekutatu). Guk erabakiko dugu nola exekutatu daitezkeen, eta horren arabera lanetan edo/eta bezeroetan hautagarri azalduko dira.
- Ingurune aldagaiak: Scripta berrerabili daitekeenez, idazten dugunean ez dakigu exekuzio garaian zein Bezero/Karpetarentzat exekutatuko den, eta kasu gehienetan

---

<sup>7</sup> [http://es.wikipedia.org/wiki/Copia\\_instant%C3%A1nea\\_de\\_volumen](http://es.wikipedia.org/wiki/Copia_instant%C3%A1nea_de_volumen)

---

informazio hau beharrezkoa izango dugu. Gure script-ean inguruneko aldagai hauek erabili ahal izango ditugu:

- ElkarBackup\_LEVEL: bere balioa JOB edo CLIENT izan daiteke
- ElkarBackup\_EVENT: bere balioa PRE edo POST izan daiteke
- ElkarBackup\_URL: bere balioa lana edo bezeroaren URL osoa da
- ElkarBackup\_ID: lanaren id kodea edo bezeroaren id kodea (zenbakia)
- ElkarBackup\_PATH: snapshot edo kopia erroaren path edo bide osoa
- ElkarBackup\_STATUS: Post-script baten irteera egoera, bestelakorik ezean beti 0 izango da

Aurrez sortutako Script baten informaziora sartzen bagara goian aipaturiko informazioaz gain, script hau non ari garen erabiltzen ikusi ahal izango dugu, deskargatzeko aukera ere emango digu.

Igotzen ditugun script hauek zerbitzariko **/var/spool/ElkarBackup/uploads** karpetan gordetzen dira.



Bezeroak Politikak Script-ak Erabiltzaileak Konfigurazioa Logak Saioa

### Script-a editatu

Deskargatu

Izena VPN\_start

Azalpena

Fitxategia Browse

Exekutatu bezero baten Pre-script moduan

Exekutatu bezero baten Post-script moduan

Exekutatu lan baten Pre-script moduan

Exekutatu lan baten Post-script moduan

Hemen erabiltzen da

[Client Cluster as pre script](#)

[Client ZamudioDB as pre script](#)

Gorde

Script baten adibidea jarri dugu ***Errepositorioa konprimitzeko scripta*** atalean.

## 7.- Aplikazioaren konfigurazioa

Menuaren **Konfigurazioa** atalean aplikazioaren hainbat parametro konfiguratzeko aukera dugu.

### 7.1.- Parametroak kudeatu

#### SSH gakoak

Gako publikoari buruz lehen ere [hitz egin dugu](#). Hemendik deskargatu dezakegu aurrez ere hemen bertan sortu dugun ElkarBackup zerbitzariaren gako publikoa.

#### MySQL zerbitzaria

Ondoren MySQL zerbitzariaren konfigurazioa kudeatzeko parametroak dauzkagu. Rsnapshot bitartez kopiatzen diren fitxategiak diskoan gordetzen dira, baina beste datu guztiak (bezero eta lanen datuak, politikak, eta abar) datu basean gordetzen dira.

MySQL zerbitzaria ElkarBackup zerbitzarian edo beste zerbitzari baten egon daiteke, beraz hemen konfiguratuko dugu datu base atzipenerako behar diren datuak.

#### Posta elektroniko bidezko mezuak

Lan bat definitzerakoan lan honek alertak bidaltzea nahi ote dugun erabaki behar dugu, zein mailako alertak bidaliko dituen eta nori.

Mezuak nori bidali

☐ Admin  
☒ Jabea  
☐ Posta helbidea

Bidaliko diren mezuak

Erroreak eta gorago

Konfigurazio lehenetsian mezuak **jabeari** bidaliko dizkio (honetan geroago sakonduko dugu), eta bidaliko dituen mezuan **Erroreak eta gorago**-koak izango dira. Jabeaz gain lan honi buruzko alertak beste norbaitek jasotzea nahiko bagenu, **Posta helbidea** laukitxoa aukeratu eta azalduko zaigun kutxan helbide elektroniko berria idatziko genuke.

Mezuak nori bidali	<input type="checkbox"/> Admin <input checked="" type="checkbox"/> Jabea <input checked="" type="checkbox"/> Posta helbidea
	<input type="text" value="helbideberria@niredomeinua.com"/>
Bidaliko diren mezuak	<input type="text" value="Errorreak eta gorago"/>

Baina zerbitzariak mezuak bidali ahal izateko bere konfigurazioa ere behar du, hau da, zerbitzariak mezu elektronikoak nola bidali behar dituen jakin behar du, eta horretarako dauzkagu ***Mailer*** parametroak.

Mailer transpor	<input type="text" value="smtp"/>
Mailer host	<input type="text" value="postazerbitzaria.niredomeinua.com"/>
Mailer user	<input type="text"/>
Mailer password	<input type="text"/>

Atal hau programatzerakoan Symfony framework-ak dituen baliabideak erabili dira, eta informazioa gehiago aurkitu daiteke [Symfony proiektuko dokumentazioan](#)<sup>8</sup>.

#### Kuota abisuak

Kuotaren gaia ere ez dugu orain arte aipatu, baina diskoaren erabileran bezero bakoitzak izango duen muga zehaztu dezakegu. Hau praktikoa izan daiteke bezero batek datu gehiegi dituenetan eta ez dugunean beste bezeroak kaltetu nahi batek leku guztia agortzen duelako.

Bezero mailan ***Kuota*** parametroan muga bat jarri eta gainditzen denean bezeroaren fitxan datu hau nabarmenduko da eta bezero eta lanen ikuspegi nagusian ere ikusiko dugu

<sup>8</sup> <http://symfony.com/doc/current/cookbook/email/email.html>

Kuota	<input type="text" value="184320"/>
Diskoaren erabilera	<div>187 MB (104%)</div>

Bezeroak					
Id-a	Izena	Diskoaren erabilera	Azken log sarrera	Egoera	Ekintzak
1	Debian Bezeroa	187 MB (104%)	2013-06-10 12:01:06 OK	Aktibo	<button>Ezabatu</button>
1.1	Debian Bezeroa/Backups karpeta	187 MB	2013-06-10 12:01:06 OK	Aktibo	<button>Ezabatu</button> <button>Restore</button>

Kontutan izan bezero batek bere ***kuota gaitzen duenean ez direla bezero horren kopiak egingo.***

Muga gaitu aurretik sistemak alerta edo abisuak bidaliko dizkigu. Parametro honen bitartez guk aukeratuko dugu noiz bidali behar duen abisua, kasu honetan bezeroaren disko erabilera kuotaren %80ra heltzen denean.

Kuota abisuen maila	<input type="text" value="80"/>	%
---------------------	---------------------------------	---

#### Beste parametro batzuk

- Aplikazioaren Log-ak zenbat denboraz mantendu behar diren automatikoki ezabatzen joan aurretik
- Aplikazioak orrialde bakoitzean erakutsiko duen lerro kopurua
- Interfazea erabiltzeko DNS izen bat erabili beharrean (adibidez <http://elkarbackup> ) IP finkoa erabiltzen dugunean (adibidez <http://IP/elkarbackup/app.php/login> ) , sistemak hainbat lekutan URLak osatzeko erabiliko duen aurrizkia (adibidearen jarraituz ***elkarbackup*** izango litzateke)

## 7.2.- Backup kokapena kudeatu

Datuak diskoan kopiatzen ari gara, eta sistemari datuak gordetzeko erabiliko duen karpeta nagusia zein izan behar den esan behar diogu. Ez badiogu besterik esaten, balio lehenetsi moduan **/var/spool/elkarbackup/backups** karpeta erabiliko du.

Kopiak egiten hasi ondoren balio hau aldatzen badugu, sistemak ez ditu datu zaharrak ezabatuko, baina restore botoiari klikatu eta datuak berreskuratzea joaten garenean ez ditu aurkituko. Kopiak egiten hasi ondoren datu hau aldatu beharra badugu, zerbitzarira kontsola bidez konektatu eta orain arteko datuak path zaharretik path berrira mugitu beharko genituzke.

Hortaz gain, elkarbackup erabiltzaile eta taldeari karpeta berrian baimenak eman beharko genieke

```
root@backups:~# chown -Rf elkarbackup.elkarbackup pathberria
```

Kontutan izan erabiliko dugun diskoa zerbitzariaren disko fisikoa izateaz gain, sare bidez muntaturiko diskoa ere izan daitekeela, adibidez [ISCSI](#)<sup>9</sup> edo [NFS](#)<sup>10</sup> protokoloen bitartez. Hau aukera interesgarria da ElkarBackup zerbitzari birtual moduan jarri nahi dugunean.

Aplikazioak backup kokapena zein izango den zehazteko aukera ematen digu, eta bi konfigurazio mota jarri ditzakegu:

- Disko lokala (Host parametroa hutsik): ElkarBackup aplikazioa instalatuta dagoen Debian zerbitzariarentzat lokala den **path** edo **bide** bat. Disko fisiko bat izan daiteke, edo aurrez beste teknikak erabiliaz muntatu zaion disko bat (adibidez iscsi bidez).
- NFS zerbitzari bateko diskoan dagoen karpeta konfiguratzeko aukera emango digu (Host parametroan zerbitzariaren izena/IPa jarrita). Hau erabili ahal izateko ElkarBackup instalazioaren Debian zerbitzariak **autofs** paketea instalatuta izan behar du (guk hasieran instalatu dugu)

<sup>9</sup> <http://es.wikipedia.org/wiki/ISCSI>

<sup>10</sup> [http://es.wikipedia.org/wiki/Network\\_File\\_System](http://es.wikipedia.org/wiki/Network_File_System)

[Bezeroak](#) [Politikak](#) [Script-ak](#) [Erabiltzaileak](#) [Konfigurazioa](#) [Logak](#) [Saioa](#)

---

### Backup kokapena kudeatu

Host

Karpeta

/var/spool/elkarbackup/backups

[Gorde](#)

Ez ahaztu kokapena aldatzen badugu Debian sistemaren **elkarbackup** erabiltzaileak baimenak izan behar dituela bertan idazteko.

### 7.3.- Biltegiaren kopia

Orain arte egin dugunarekin hainbat zerbitzarietan dauzkagun datuen kopia egitea lortu dugu, diskoan ahalik eta leku gutxien okupatzen eta politika ezberdinak erabiliaz.

Baina datu hauek disko batean dauzkagu, eta disko horri zerbait gertatuko balitzaio, agur gure kopiak.

Garrantzitsua litzateke kopia guztiak gordetzen dituen disko horren **beste kopia bat** beste nonbaiten izatea, **fisikoki beste leku batean** (hondamendi egoera batean kaltetua ez izateko), eta posible den heinean bigarren kopia hori gure kopia nagusiarekin era automatikoan sinkronizatuta izatea.

Menuan **Konfigurazioa** → **Biltegiaren kopia egiteko script-a** aukeran **Backup script deskargatu** aukera ikus dezakegu.

[Bezeroak](#) [Politikak](#) [Script-ak](#) [Erabiltzaileak](#) [Konfigurazioa](#)

---

### Biltegiaren kopia egiteko script-a

[Backup script](#) [Deskargatu](#)

Botoi horren gainean klikatzen badugu, ondorengo script hau deskargatuko dugu:

```
#!/bin/bash

MYSQL_DB=ElkarBackup
MYSQL_HOST=localhost
MYSQL_PASSWORD=root
MYSQL_USER=root
REPOSITORY=/var/spool/ElkarBackup/backups
SERVER=ElkarBackup
SERVER_USER=ElkarBackup
UPLOADS=/var/spool/ElkarBackup/uploads

ssh "$SERVER_USER@$SERVER" "cd '$REPOSITORY'; find . -maxdepth 2 -mindepth 2" |
sed s/^..// | while read jobId
do
    echo Backing up job $jobId
    mkdir -p $jobId 2>/dev/null
    rsync -aH --delete "$SERVER_USER@$SERVER:$REPOSITORY/$jobId/" $jobId
done
echo Backing up mysql DB
ssh "$SERVER_USER@$SERVER" "mysqldump -u$MYSQL_USER -p$MYSQL_PASSWORD
-h$MYSQL_HOST $MYSQL_DB" > ElkarBackup.sql
echo Backing up uploads
rsync -aH --delete "$SERVER_USER@$SERVER": "$UPLOADS/" uploads
```

Script hau beste makina batean (aurrerantzean **Morroia** deituko diogu) exekutatzen badugu:

1. ElkarBackup paketea instalatuta dagoen zerbitzarira konektatuko da eta hor dauzkagun kopia guztien sinkronizazioa abiatuko du.
2. Ondoren MySQL datu basearen backup bat egin eta ElkarBackup.sql fitxategian datu guztiak jasoko ditu
3. Azkenik **/var/spool/ElkarBackup/uploads** karpetara igo diren script guztiak ere kopiatuko ditu.

Beraz nahikoa izango genuke script honen exekuzioa **Morroia** zerbitzarian guk nahi dugun egun eta orduan programatzea biltegiaren kopia osoa bere diskoan izateko.

Esan beharrik ez dago, baina hondamendi egoera baten aurrean ez luke zentzu handirik izango bi zerbitzariak leku fisiko berdinean izateak .....

## Automatizazioa

Exekuzio hau automatizatu nahi badugu, hau da, **Morroia** zerbitzariak programaturiko egunean eta orduan **ElkarBackup** zerbitzariko biltegia automatikoki sinkronizatzea nahi badugu, ezinbestekoa izango zaigu berriro ere gako publiko/pribatuaren estrategia erabiltzea.

Kasu honetan **Morroia** zerbitzariko gako publikoa **ElkarBackup** zerbitzarian inportatu beharko dugu, edo hobeto esanda, **Authorized keys** fitxategian gehitu beharko dugu. Hau egiteko **Gakoa gehitu** botoiaren gainean klikatuko dugu.

Ikusten dugunez bi eremu dira bete behar ditugunak:

- Iruzkina: Gako identifikatzeko erabiliko dugun izena
- Gakoa: Morroia zerbitzariko **gako publikoa**

Gauzak errazteko, **Morroia** zerbitzarian elkarbackup izeneko erabiltzaile bat sortuko dugu

```
root@morroia:~$ adduser elkarbackup
'elkarbackup' erabiltzailea gehitzen...
'elkarbackup' (1001) talde berria gehitzen...
'elkarbackup' (1001) erabiltzaile berria 'elkarbackup' taldearekin gehitzen...
'/home/elkarbackup' karpeta nagusia sortzen...
'/etc/skel'(e)tik fitxategiak kopiaitzen...
UNIX-pasahitz berria sartu:
UNIX-pasahitz berria sartu berriro:
passwd: pasahitza ongi eguneratu da
elkarbackup(r)en erabiltzaile informazioa aldatzen
Idatzi balio berria, edo sakatu 'Sartu' tekla lehenetsirako
    Izen osoa []:
    Gela zenbakia []:
    Laneko telefonoa []:
    Etxeko telefonoa []:
    Bestelakoa []:
Informazioa zuzena da? [B/e] B
```

Ondoren erabiltzaile horrekin saioa ireki

```
root@morroia:~$ su - elkarbackup
```

eta RSA gakoa sortuko dugu

```
elkarbackup@morroia:~$ ssh-keygen -t rsa
```



```

Generating public/private rsa key pair.
Enter file in which to save the key (/home/elkarbackup/.ssh/id_rsa):
Created directory '/home/elkarbackup/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/elkarbackup/.ssh/id_rsa.
Your public key has been saved in /home/elkarbackup/.ssh/id_rsa.pub.
The key fingerprint is:
1a:65:fd:52:08:05:89:8a:3a:89:68:23:72:c3:37:0e elkarbackup@portatil59
The key's randomart image is:
+--[ RSA 2048 ]-----+
|      .o+.      |
|      . .o .    |
|    . . o o .   |
|    . . o o     |
|o+    . S . .   |
|OoE o o .      |
|+o.= ..        |
|      .         |
|               |
+-----+

```

Orain gako publikoa zein den ikusi eta kopiatu egingo dugu

```

elkarbackup@morroia:~$ cat .ssh/id_rsa.pub
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQACuklE6TI16fU5pmTeU4APrSgG24eblwQdnHNtntUIIRSyAkAe
mPil2GDpufXgPKqT+FQV02z4JiaoTMMhQpsGqS/Shz/KE/MA7pm8k9v6qnFKVpY6HXZZyvgYhH+Yy6F
xxDk+QGQqQMnabzmanyxcBBdQ3ZdluYbwT5kdIgAJR8eTLN/M08hrKKKeQGbEVXP3GCPWYsiDV2p6VgR
gkPzCSWUgMP63668ZAoNq8mlhW8RF+BEYDF9TPH7PJaEhc+Ea5LiiggD/E2lqQmFGYTqbjELKT4b97y
6nDj+2UIG0pvqZ/dN0ZMdsCMX577e8ppkafMqgwGT3D7Af4gD9KninL3 ElkarBackup@morroia

```

eta ElkarBackup web aplikazioaren interfazeak ematen digun aukera aprobetxatuz, gakoa hau gehituko dugu zerbitzariko **authorized\_keys** fitxategian.

### Authorized keys

Comment / Key		
	<input type="text" value="Morroia zerbitzaria"/>	
	<input type="text" value="ssh-rsa AAAAB3NzaC1yc2EAAA"/>	<input type="button" value="Ezabatu"/>
		<input type="button" value="Gorde"/> <input type="button" value="Gakoa gehitu"/>

Gorde botoiari eman eta nahiko.

Ondoren Morroia zerbitzarian karpeta bat sortuko dugu erreplika bertan izateko, eta bertara kopiatuko dugu web interfazetik deskargatu dugun scripta.

```
ElkarBackup@morroia:~$ ls -la biltegiarenkopia/  
guztira 12  
drwxrwxr-x 2 elkarbackup elkarbackup 4096 eka 10 17:02 .  
drwxr-xr-x 4 elkarbackup elkarbackup 4096 eka 10 16:53 ..  
-rwxrwxr-x 1 elkarbackup elkarbackup 695 eka 10 15:24 copyrepository.sh
```

eta exekuzioa testatuko dugu

```
elkarbackup@portatil59:~$ cd biltegiarenkopia/  
elkarbackup@portatil59:~/biltegiarenkopia$ ./copyrepository.sh  
Backing up job 0002/0002  
Backing up job 0001/0001  
Backing up mysql DB  
Backing up uploads
```

Ikusten dugunez erreplikazioa egiten du inongo erabiltzaile eta pasahitzik eskatu gabe.

```
elkarbackup@portatil59:~/biltegiarenkopia$ ls -la  
guztira 128  
drwxrwxr-x 5 elkarbackup elkarbackup 4096 eka 10 17:04 .  
drwxr-xr-x 4 elkarbackup elkarbackup 4096 eka 10 16:53 ..  
drwxrwxr-x 3 elkarbackup elkarbackup 4096 eka 10 17:04 0001  
drwxrwxr-x 3 elkarbackup elkarbackup 4096 eka 10 17:04 0002  
-rwxrwxr-x 1 elkarbackup elkarbackup 695 eka 10 15:24 copyrepository.sh  
-rw-rw-r-- 1 elkarbackup elkarbackup 103785 eka 10 17:04 ElkarBackup.sql
```



---

drwxr-xr-x 2 elkarbackup elkarbackup 4096 eka 6 15:32 uploads
---

## 8.- Oinarrien errepeaso txiki bat: Rsnapshot

Aurrez aipatu bezala, ElkarBackup rsync eta batez ere rsnapshot tresna libreetan oinarritzen den aplikazioa da, eta gure aplikazioak erabiliko duen logika hein handi batean rsnapshot-ek erabiltzen duen berdina denez, merezi du hau pixkat bada ere azaltzea.

Rsnapshot tresnak bere oinarrian kontzeptu hauek erabiltzen ditu:

- Kopiatu behar diren datuen jatorria: Hau da, nondik kopiatu behar ditu datuak ?
- Maiztasuna: Zenbatero kopiatu behar ditu? Egunean zehar guk esandako orduetan, eguneroko kopia guk zehaztutako asteko egunetan, hileroko kopiak guk zehaztutako hileko egunean ....
- Erretentzioa: Maiztasun bakoitzeko zenbat kopia mantendu behar ditu ? Hau da, agian egunean zehar egindako 4 kopia mantentzea nahiko da, baina hilekoak agian bi urtez mantendu nahiko genituzke.

Ondoren azalduko ditugun puntuak asko lagunduko digute logika hau ulertzen.

### 8.1.- Maiztasuna

Kopiak zenbatero egin behar dituen adierazi beharko diogu, aukerak honako hauek direlarik:

- Egunean zehar (Hourly): Ordu ezberdinetan **egingo dituen** kopiak zehaztuko ditugu. Sistemak azpitik Hourly.0, Hourly.1, Hourly.2 ..... izendatuko ditu.
- Egunero (Daily): Asteko egun bakoitzean **egingo dituen** kopiak zehaztuko ditugu, zein astegun desberdinetan eta zein ordutan egingo dituen zehaztuta. Sistemak azpitik Daily.0, Daily.1, Daily.2 ..... izendatuko ditu.
- Astero (Weekly): Astean behin **egingo dituen** kopiak zehaztuko ditugu, zein astegunetan (bakarra) eta zein ordutan egingo dituen zehaztuta. Sistemak azpitik Weekly.0, Weekly.1, Weekly.2 ..... izendatuko ditu.
- Hilero (Monthly): Hilero **egingo dituen** kopiak zehaztuko ditugu, hileko zein egunean eta zein ordutan egingo dituen zehaztuta. Sistemak azpitik Monthly.0, Monthly.1, Monthly.2 ..... izendatuko ditu.

Goian **egingo dituen** esan dudanean, ez dut egi borobila esan. Batzuetan kopia egingo du, eta beste batzuetan aurrez dauden kopien errotaioa egingo du, eta hau nola egiten duen berehala azalduko dugu, errotaioa azaltzerakoan.

## 8.2.- Erretentzioa

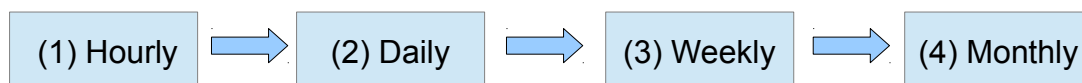
Erretentzioarekin maiztasun bakoitzean **zenbat karpeta** gorde behar dituen adierazten diogu. Adibidez:

- Egunean zehar (Hourly) egiten dituen kopietan 4ko erretentzioa jartzen badiogu, horrek esan nahi du berak Hourly.0, Hourly.1, Hourly.2 eta Hourly.3 karpetetaz azken 4 kopiak mantenduko dituela. Hourly.0 karpetan datu berrienak, eta Hourly.3 karpetan datu zaharrenak.
- Egunero (Daily) egiten dituen kopietan 5eko erretentzioa jartzen badiogu, horrek esan nahi du berak Daily.0, Daily.1, Daily.2, Daily.3 eta Daily.4 karpetetaz azken 5 egunetako kopiak mantenduko dituela. Daily.0 karpetan datu berrienak, eta Daily.3 karpetan datu zaharrenak.

Logika berdina erabiltzen du astero (Weekly) eta hilero (Monthly) ataletan adierazitako erretentzioarekin.

## 8.3.- Errotazioa

Hau ez da zaila, baina ondo ulertu behar da. Maiztasuna definitzerakoan hauen arteko mailakatzea ere definitzen ari gara, nolabait Monthly-aren **azpitik** Weekly-a dagoela esanez, Weekly-aren **azpitik** Daily-a, eta azkenik Daily-aren **azpitik** Hourly-a. Hourly-aren azpitik ez dago beste mailarik.



Kopia politika ezberdinak izan ditzakegu, baina adibidea azaltzeko demagun gure programazioa honela osatzen dela:

- Hourly-ak 11:00, 14:00 eta 16:00etan, 3ko erretentzioarekin
- Daily-ak astelehenetik ostiralera 21:00etan, 10eko erretentzioarekin
- Weekly-ak larunbatetan 21:00etan, 5eko erretentzioarekin
- Monthly-ak hilaren batean 21:00etan, 24ko erretentzioarekin

Kasu honetan kopiak honela egingo lirateke:

- Hourly-a exekutzeko ordua heltzen denean, azkena (Hourly.2) ezabatu egiten da, eta besteen izenak aldatzen dira. Hourly.1 → Hourly.2 eta Hourly.0 → Hourly.1. Une horretan kopia berria egiten du Hourly.0 karpetan.

- Daily-a exekutatzeko ordua heltzen denean, azkena (Daily.9) ezabatu egiten da, eta besteen izenak aldatzen dira. Daily.8 → Daily.9, Daily.7 → Daily.8, ..., Daily.0 → Daily.1 . Desberdintasuna hemen dator, kasu honetan **ez da kopia berri bat egingo** Daily.0 sortzeko, aurreko mailako azken kopia erabiliz errotazio baten bitartez osatuko da: Hourly.2 → Daily.0
- Weekly-a exekutatzeko ordua heltzen denean, azkena (Weekly.4) ezabatu egiten da, eta besteen izenak aldatzen dira. Weekly.3 → Weekly.4, Weekly.2 → Weekly.3, ..., Weekly.0 → Weekly.1 . Kasu honetan ere **ez da kopia berri bat egingo** Weekly.0 sortzeko, aurreko mailako azken kopia erabiliz errotazio baten bitartez osatuko da: Daily.9 → Weekly.0
- Monthly-a exekutatzeko ordua heltzen denean, azkena (Monthly.23) ezabatu egiten da, eta besteen izenak aldatzen dira. Monthly.22 → Monthly.23, Monthly.21 → Monthly.22, ..., Monthly.0 → Monthly.1 . Kasu honetan ere **ez da kopia berri bat egingo** Monthly.0 sortzeko, aurreko mailako azken kopia erabiliz errotazio baten bitartez osatuko da: Weekly.4 → Monthly.0

Ulertu behar dena hau da: Errotazio batek **beheragoko mailako azken kopia bakarrik** mugi lezake, inoiz ere ez bestek. Hau da:

- Hourly erretentzioa 3 bada, Dailyk Hourly.2 ukituko du, inoiz ez bestek.
- Hourly erretentzioa 2 bada, Dailyk Hourly.1 ukituko du, inoiz ez hourly.0

Logika guzti honetan argi izan behar dugun muga bat dago. Inoiz ez dira ukitzen .0 bukatzen diren karpetak, eta horregatik **erretentzioa ezin da 1 izan goragoko mailarik dagoenean** , Daily-k ez du inoiz hourly.0 ukituko, ezta Weekly-k daily.0 ere.

Hau guztia ulertzea garrantzitsua da, gero web interfazeaz politikak programatzerakoan azpitik aplikatuko den logika hau delako.

## 9.- Eranskinak

### 9.1.- Irudia deskargatu ondoren

Kontutan izan behar dugu irudi hau deskargatzen dugun guztiok gako berdina erabiltzen ari garela, beraz honek segurtasun arazo bat suposatu lezake.

Web interfazetik **Konfigurazioa** → **Parametroak kudeatu** atalean **Deskargatu** botoia agertzen zaigu sistemak detektatzen duelako gakoa aurretik sortua izan dela. Gako hori azpitik ezabatzen badugu, **Deskargatu** beharrean **Sortu** agertuko zaigu, beraz gako berria sortzeko aukera izango dugu.

```
root@ElkarBackup:~# rm /var/lib/elkarbackup/.ssh/id_rsa.pub
```

### 9.2.- Errepositorioa konprimitzeko scripta

Demagun erabiltzaile ezberdin batzuk fitxategi berdinen kopia ezberdinak gordetzen dituztela sarean, bakoitzak bere karpetan kopia bat duelarik. Dokumentu hauek ofimatikoak direnean (normalean ez oso handiak) ez da hain arazo handia izaten, baina beste mota batzuetakoak direnean, bideoak, software eguneraketak, eta abar, sarean leku asko jaten dute, informazio berdina delarik.

Badakigu kopia egin ondoren datu hauek segurtasun kopia mantentzen duen diskoan ez direla aldatuko, beraz aukera izan dezakegu diskoan behin bakarrik gorde eta beste kopia guztiak hardlink bidez estekatzeko.

Adibiderako, Debian bezeroan daukagun fitxategiren bat kopiatu eta beste izen batekin itsatsiko dugu **/media/Backups** karpetan. Fitxategi berdina den arren, bi aldiz dago eta diskoan leku bikoitza ari da okupatzen.

```
root@DebianBezerao:~# cd /media/Backups/Software/
root@DebianBezerao:/media/Backups/Software# cp vlc-2.0.6-win32.exe vlc-2.0.6-win32-kopia.exe
root@DebianBezerao:/media/Backups/Software# ls -lah
total 209M
drwxr-xr-x 2 root root 4,0K jun 12 12:44 .
drwxr-xr-x 4 root root 4,0K jun  7 11:24 ..
-rw-r--r-- 1 root root 1,1M nov 18 2010 7z920.exe
-rw-r--r-- 1 root root 164M may  3 18:16 LibreOffice_4.0.3_Linux_x86_deb.tar.gz
-rw-r--r-- 1 root root 22M abr 15 19:11 vlc-2.0.6-win32.exe
-rw-r--r-- 1 root root 22M jun 12 12:44 vlc-2.0.6-win32-kopia.exe
```

Lan honen kopia egiteko ardura duen lanean sartu eta **Exekutatu orain** botoiaren bidez

abiatuko dugu. Lana bukatzen denean ikusiko dugu ElkarBackup zerbitzariko karpetan fitxategi biak daudela.

```
#cd /var/spool/elkarbackup/backups/0001/0001/Hourly.0/media/Backups/Software/
# ls -lah
total 209M
drwxrwxr-x 2 elkarbackup elkarbackup 4,0K jun 12 12:44 .
drwxrwxr-x 4 elkarbackup elkarbackup 4,0K jun  7 11:24 ..
-rw-rw-r-- 3 elkarbackup elkarbackup 1,1M nov 18 2010 7z920.exe
-rw-rw-r-- 3 elkarbackup elkarbackup 164M may  3 18:16 LibreOffice_4.0.3_Linux_x86_deb.tar.gz
-rw-rw-r-- 3 elkarbackup elkarbackup 22M abr 15 19:11 vlc-2.0.6-win32.exe
-rw-rw-r-- 2 elkarbackup elkarbackup 22M jun 12 12:44 vlc-2.0.6-win32-kopia.exe
```

eta ikusten dugu ere ez daudela hardlink moduan estekatuta, *inode* ezberdina dutelako

```
# ls -lahi
total 209M
40831 drwxrwxr-x 2 elkarbackup elkarbackup 4,0K jun 12 12:44 .
40828 drwxrwxr-x 4 elkarbackup elkarbackup 4,0K jun  7 11:24 ..
29332 -rw-rw-r-- 3 elkarbackup elkarbackup 1,1M nov 18 2010 7z920.exe
29333 -rw-rw-r-- 3 elkarbackup elkarbackup 164M may  3 18:16 LibreOffice_4.0.3_Linux_x86_deb.tar.gz
29334 -rw-rw-r-- 3 elkarbackup elkarbackup 22M abr 15 19:11 vlc-2.0.6-win32.exe
74101 -rw-rw-r-- 2 elkarbackup elkarbackup 22M jun 12 12:44 vlc-2.0.6-win32-kopia.exe
```

Orain ikusiko dugu nola erabili dezakegun postscript script bat hau konpontzeko.

Script honek [Hash](#)<sup>11</sup> berdina duten fitxategiak bilatu eta hardlink bidez estekatuko ditu

```
#!/bin/bash
# lehenengo tamainaz konparatu eta errepikatuta ez daudenak deskartatu

cd $ELKARBACKUP_PATH
lastHash=''
lastFile=''

find . -mount -type f -printf '%15s %p\0'|sort -nrz|uniq -zDw15|tr "\0" "\n"|
cut -b17- |tr "\n" "\0"|xargs$
do
    if [ "x$lastHash" = "x$currentHash" ]
    then
        rm "$file"
        ln "$lastFile" "$file"
```

<sup>11</sup> <https://eu.wikipedia.org/wiki/Hashing> / [https://es.wikipedia.org/wiki/Funci%C3%B3n\\_hash](https://es.wikipedia.org/wiki/Funci%C3%B3n_hash)



```
fi
lastHash=$currentHash
lastFile="$file"
done
```

Scripta igoko dugu:

- Izena: Errepositorioa txikitu
- Lanen PostScript moduan erabiltzeko baimendu

Orain Debian bezeroaren lana editatu eta script hau **PostScript** bezala aukeratuko dugu, eta azkenik lanaren exekuzioa abiatuko dugu.

```
# ls -lahi
total 209M
40838 drwxrwxr-x 2 elkarbackup elkarbackup 4,0K jun 12 15:22 .
40835 drwxrwxr-x 4 elkarbackup elkarbackup 4,0K jun  7 11:24 ..
29332 -rw-rw-r-- 4 elkarbackup elkarbackup 1,1M nov 18  2010 7z920.exe
29333 -rw-rw-r-- 4 elkarbackup elkarbackup 164M may  3 18:16 LibreOffice_4.0.3_Linux_x86_deb.tar.gz
29334 -rw-rw-r-- 8 elkarbackup elkarbackup  22M abr 15 19:11 vlc-2.0.6-win32.exe
29334 -rw-rw-r-- 8 elkarbackup elkarbackup  22M abr 15 19:11 vlc-2.0.6-win32-kopia.exe
```

Orain lehenengo zutabeen agertzen diren inode zenbakietan erreparatzen badugu, konturatuko gara kopiak diren fitxategiak inode berdinarekin daudela, beraz diskoaren leku berdinerara apuntatzen ari dira eta ez dute erabilitako espazioa bikoizten.

### 9.3.- Lanak ordenatu

Politika berdinean dauden lan guztiak bata bestearen ondoren exekutatzen dira. Lanen batek exekutatzen denbora asko behar badu atzetik daudenen exekuzioa nabarmen atzeratzea posible da. Hau gertatzeko bi kasu detektatu dira:

- Hasierako kopia egitean informazio asko transmititu behar bada.
- Lan batek fitxategi asko badauzka (probetan 302000 fitxategi).

Lehenengo kasua ez da kezkgarria puntuala delako. Bigarren kasuan berriz, rsync-ek direktorio zuhaitz handi bat osorik irakurri behar du bai bezeroan eta baita zerbitzarian ere, eta horrek denbora eraman dezake. Gauza bera gertatzen da berriz ere behe mailako errotazioek eskatzen duten "cp -al" eragiketekin. Gero berriz ere disko erabilera kalkulatzeko orduan.

Kasu hauetan, agian lan batzuek besteek baino lehenago exekutatzea interesatzen zaigu, eta horretarako **lanei lehentasunak esleitzeko** aukera jarri da, Menuaren **Bezeroak → Lanak ordenatu** sarreraren bitartez.

Lehentasunak esleitzeko lan guztiak zerrenda batean erakutsiko dira eta zerrenda hori ordenatzeko aukera dago. Zerrendan duten ordenean egingo dira segurtasun kopiak.

#### 9.4.- Irudia beste virtualizazio sistema batzuetara bihurtzen

Erabiltzen duzun virtualizazio sistemak Raw formatua onartzen badu ez duzu lan askorik izango, bestelakoetan irudia formatu batetik bestera pasatu beharko duzu.

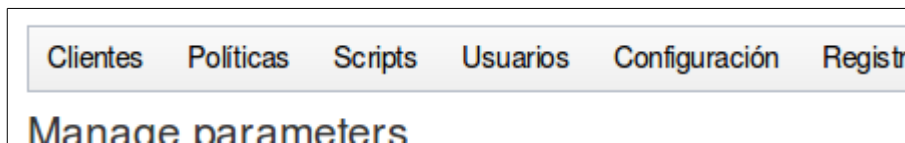
Vmware erabiltzen baduzu, hemen duzu hau nola egiten den ikusteko laguntza: <http://wiki.laptop.org/go/VMware/Convert>

```
:~# qemu-img convert ElkarBackupServerBase2GB1.0.9.img -O vmdk
ElkarBackupServerBase2GB1.0.9.vmdk
```

VirtualBox-erako beste hau

```
:~# Deskargak$ VBoxManage convertfromraw ElkarBackupServerBase2GB1.0.9.img
ElkarBackupServerBase2GB1.0.9.vdi
```

#### 9.5.- Hodeiarekin sinkronizazioa



Gaur egun segurtasun kopiak kudeatzeko izango den soluzio bat garatzerakoan ezinbestekoa da hodeian dauden zerbitzuekin sinkronizatzeko aukerak jartzea merezi ote duen pentsatzea.

Aplikazioan ez da helburu honekin ezer garatu ikusi dugulako zerbitzu hauek ematen dituzten aukerekin ezer garatu beharrik ez dagoela.

Dropbox ezagunera joaten bagara, Linux zerbitzari batean [daemon moduan exekutatze](#)<sup>12</sup> moduko instalazioa duela ikus daiteke.

Elkarbackup zerbitzarian Dropbox instalatuta badugu, Postscript bat garatu dezakegu lan bakoitzean Dropbox karpetara estekatu behar diren fitxategiak hardlink bidez kopiatzeko, eta horrela hodeian guk nahi ditugun datuen kopia ere izango genuke.

Eztabaida polit bat abiatu daiteke hau komenigarria den ala ez erabakitzeko, datuen pribatutasuna eta horrelakoak tartean direlarik, baina bueno, hori beste kontu bat da.

#### 9.6.- Windows bezeroen kopiak eta snapshot-ak

Oso ezaguna ez bada ere, Windows sistemak aspalditik snapshot-ak egiteko aukera ematen dute. Tresna horri esker dauka Windows-ek sistemaren egoera automatikoki gordetzeko egoera, seguru behin baino gehiagotan erabili izan duzula arazo baten

<sup>12</sup> <http://pablo.sarubbi.com.ar/sysadmin/instalar-dropbox-desde-una-terminal-de-linux>

---

aurrean aurreko astean sistemak zuen egoerara bueltatzeko.

[Bukatzeke]

## 9.7.- NFS urruneko diskoa

[Egiteke]

## 10.- Lizentzia

- Egilea: Pedro Arreitunandia Ituarte
- Data: 2013/06/12
- Lizentzia: Creative Commons BY-SA: Aitortu – Partekatu berdin:

<http://creativecommons.org/licenses/by-sa/2.5/es/>

