Linux Disaster Recovery best practices with rear

Gratien D'haese
IT3 Consultants

Who am I

- Independent Unix System Engineer since 1996
- Unix user since 1986
- Linux user since 1991
- Open Source contributor:
  - Make CD-ROM Recovery (mkCDrec)
  - Relax and Recover (rear)
  - SIM Installation and Logging (similar)
  - Adhocracy (adhocr)

Disaster Recovery

- Business Continuity Planning
  - A business continuity plan specifies how a company plans to restore core business operations when disasters occur
- Disaster Recovery
  - Disaster recovery looks specifically at the technical aspects of how a company can get back into operation using backup facilities

Disaster Recovery Concerns

- Uptime
  - Quick restores with minimal or no manual steps after the recovery
- Reliability
  - Avoid corrupted file systems and that system boots after recovery
- Cost
  - DR solutions need to be affordable
- Complexity
  - DR plans tend to be too complex.

Getting started with Relax and Recover (rear)

- Download it from
  - The official tar-balls
    - https://github.com/rear/rear/downloads/
  - The rear-snapshot rpm's build from Github
    - http://download.opensuse.org/repositories/Archiving:/Backup:/Rear/
  - The official source
    - https://github.com/rear/rear
  - The official repo's (Fedora, EPEL and SLES)
    - yum install rear
    - zypper install rear

Installation of rear

- E.g. on Fedora 17

```
# yum install rear
```

```
Installing:
 rear                           noarch          1.13.0-1.fc17                fedora          327 k
Installing for dependencies:
 at                             i686            3.1.13-7.fc17                fedora           61 k
 bc                             i686            1.06.95-6.fc17               fedora          106 k
 binutils                       i686            2.22.52.0.1-5.fc17           fedora          3.6 M
 ed                             i686            1.5-3.fc17                   fedora           72 k
 ethtool                        i686            2:3.2-2.fc17                 fedora           93 k
 genisoimage                    i686            1.1.11-10.fc17               fedora          338 k
 ….
Install  1 Package (+40 Dependent packages)
Total download size: 21 M
Installed size: 65 M
Is this ok [y/N]: y
```

- **We also need syslinux (and to boot on USB: extlinux)**
  `# yum install syslinux`
- **Install nfs-utils, cifs-utils, rsync if required**
- **Do not forget openssh(-clients)**

Decide on DR strategy

- Which backup mechanism to use?
  - GNU tar, rsync, bacula, commercial backup program
- Where will the backups reside?
  - NFS share, CIFS share, external USB disk, tape, local spare disk
  - Remote network location
- How shall we start the rescue image
  - Via CDROM (ISO image), tape (OBDR), network (PXE), USB disk

Backup Types

- The major backup types available are
  - **NETFS**: NFS, CIFS, USB, TAPE
  - RSYNC: rsync method
  - REQUESTRESTORE, EXTERNAL
  - BACULA (open source backup software)
  - DP, NBU, TSM, GALAXY[7] (commercial stuff)
- Some not (yet) implemented backup types (waiting on sponsors)
  - NSR (Legato Networker)
  - CDROM

BACKUP type NETFS

isolinux

OUTPUT=ISO

pxelinux

network

OUTPUT=PXE
BACKUP=NETFS

(NFS|CIFS|local)
disks

OUTPUT=ISO
BACKUP=NETFS

Tape drive

OUTPUT=OBDR
BACKUP=NETFS

extlinux

External USB disks

OUTPUT=USB
BACKUP=NETFS

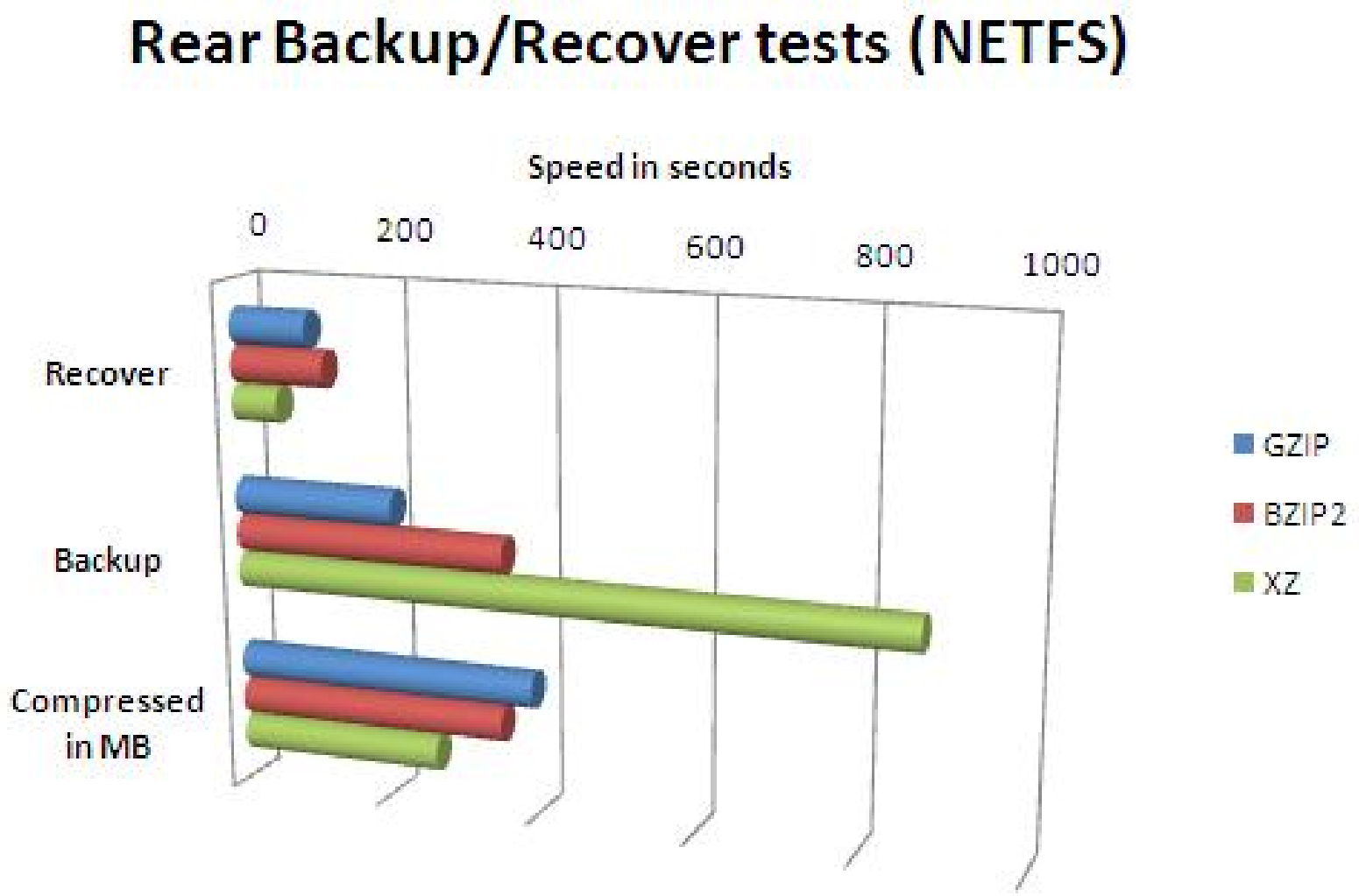
Location BACKUP_URL

- **BACKUP=NETFS**
- **BACKUP_URL** can be
  - File type: BACKUP_URL=file:///directory/
  - NFS type: BACKUP_URL=nfs://nfs-server/directory/
  - CIFS type: BACKUP_URL=cifs://samba/directory/
  - USB type: BACKUP_URL=usb:///dev/sdc1/directory/
  - Tape type: BACKUP_URL=tape:///dev/nst0

Backup Program

- BACKUP=NETFS
- /usr/share/rear/conf/default.conf
  - By default is BACKUP_PROG=tar
  - However, BACKUP_PROG=rsync is possible for local attached storage
  - BACKUP_PROG_COMPRESS_OPTIONS="--gzip"
  - BACKUP_PROG_COMPRESS_SUFFIX=".gz"
  - BACKUP_PROG_EXCLUDE=('/tmp/*' '/dev/shm/*')

BACKUP_PROG_COMPRESS_OPTIONS



/etc/rear/local.conf

- Define your settings in /etc/rear/local.conf (or /etc/rear/site.conf)
- # grep -v -E '(^#|^$)' /etc/rear/local.conf
  OUTPUT=ISO
  MODULES_LOAD=(vmxnet)
- Add:
  BACKUP=NETFS
  BACKUP_URL=nfs://server/path
- On NFS server backup => /path/$(hostname)/

Rear dump

- View system configuration:

```
# rear dump
Relax and Recover 1.13.0 / $Date$
Dumping out configuration and system information
This is a 'Linux-x86_64' system, compatible with 'Linux-i386'.
System definition:
                                    ARCH = Linux-i386
                                      OS = GNU/Linux
                          OS_MASTER_VENDOR =
                         OS_MASTER_VERSION =
                     OS_MASTER_VENDOR_ARCH =
                  OS_MASTER_VENDOR_VERSION =
       OS_ MASTER_VENDOR_VERSION_ARCH =
                                OS_VENDOR = Fedora
                                OS_VERSION = 16
                           OS_VENDOR_ARCH = Fedora/i386
                        OS_VENDOR_VERSION = Fedora/16
```

Rear help

- Usage: rear [-dDsSvV] [-r KERNEL] COMMAND [--ARGS...]
- Available options:
  - **-d** debug mode; log debug messages
  - **-D** debugscript mode; log every function call
  - **-r KERNEL** kernel version to use; current: '2.6.42.3-2.fc15.i686.PAE'
  - **-s** simulation mode; show what scripts rear would include
  - **-S** step-by-step mode; acknowledge each script individually
  - **-v** verbose mode; show more output
  - **-V** version information

Rear help

- Usage: rear [-dDsSvV] [-r KERNEL] COMMAND [--ARGS...]
- List of commands:
  - checklayout — check if the disk layout has changed
  - format — format and label media for use with rear
  - mkbackup — create rescue media and backup system
  - mkbackuponly — backup system without creating rescue media
  - mkrescue — create rescue media only
  - recover — recover the system; only valid during rescue
  - savelayout — save the disk layout of the system
  - shell — start a bash within rear; development tool

Disaster Recovery in Practice

- Gather system information
- Store the disk layout
  - Partitioning, LVM and RAID configuration
  - File systems, file system labels ...
  - Boot loader (GRUB, LILO, ELILO)
- Make a system backup (OS and user data)
- Create boot-able rescue media with system configuration (and optional with backup data)
- All steps are done “**online**”

Rear mkrescue

- Will create an ISO image stored as
  - /tmp/rear-$(hostname).iso
  - On NFS server as /path/$(hostname)/rear-\ $(hostname).iso
- Inspect file /var/lib/rear/layout/disklayout.conf
- Try to boot from the ISO image into the RESCUE system
  - Use 'dmesg' to check if devices were found

Rear mkbackup

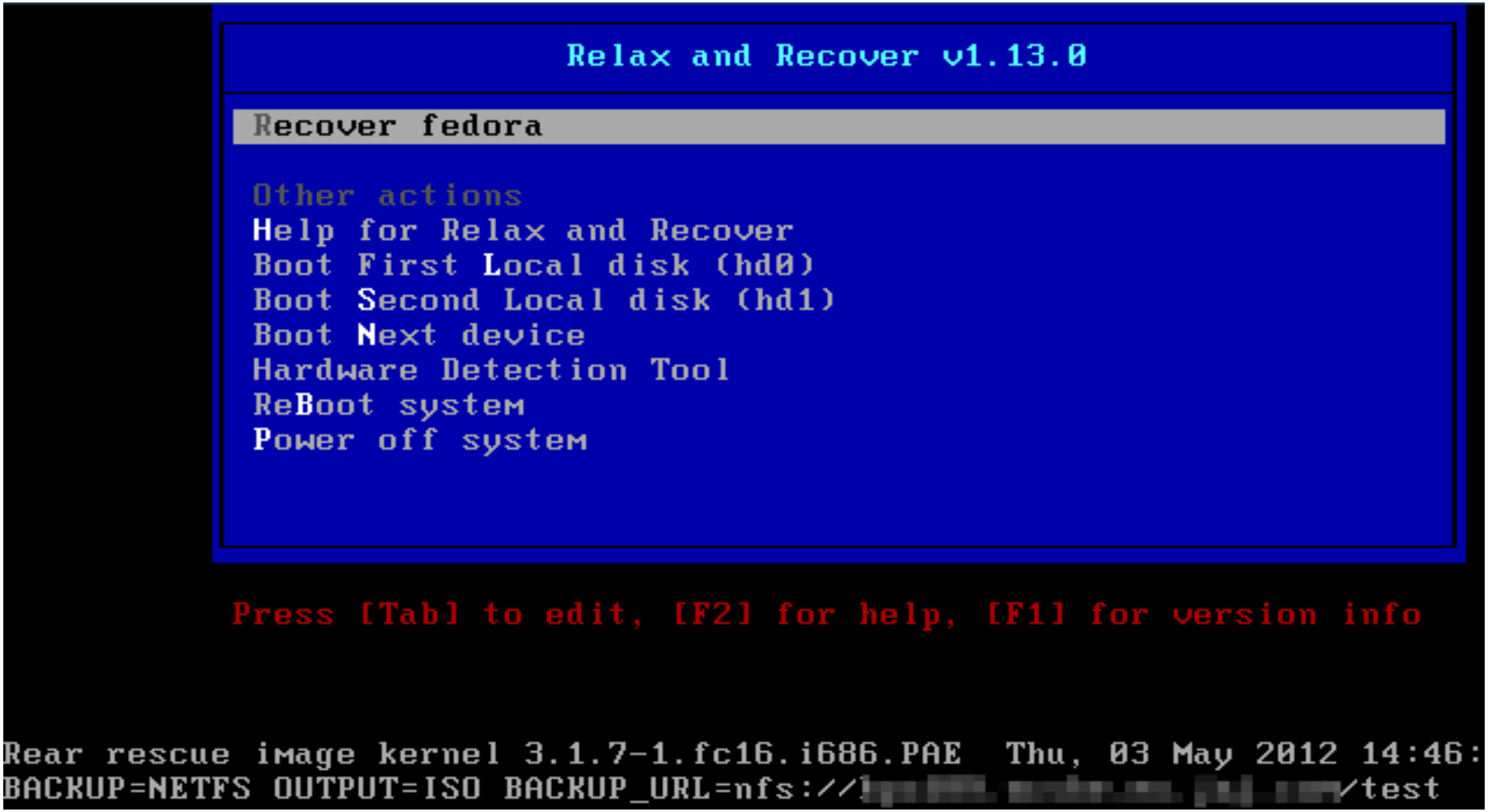
- Create rescue image with backup archive
- Do not forget to browse through the `/tmp/rear-$(hostname).log` file for errors

Recovery Process in detail

- Boot system from rescue media
- Restore disk layout
  - Create partitions, RAID configuration and LVM
  - Create file systems (mkfs, mkswap)
  - Configure file systems (labels, mount points)
- Restore the backup data
- Restore the boot loader
- Inspect & Reboot

Recover with rear (1)

- Boot rescue image and select 'recover'



Recover with rear (2)

- Wait until you see the login prompt

```
Attempting to start the DHCP client daemon
Running 60-network-devices.sh...
Running 62-routing.sh...
* * * Rescue System is ready * * *



Relax and Recover 1.13.0 / $Date$

Relax and Recover comes with ABSOLUTELY NO WARRANTY; for details see
the GNU General Public License at: http://www.gnu.org/licenses/gpl.html

Host fedora using Backup NETFS and Output ISO
Build date: Thu, 03 May 2012 14:45:18 +0200


Fedora release 16 (Verne)
Kernel 3.1.7-1.fc16.i686.PAE on an i686 (tty1)

fedora login: root

Welcome to Relax and Recover. Run "rear recover" to restore your system

RESCUE fedora:~ # _
```

Recover with rear (3)

```
RESCUE fedora:~ # rear -v recover
Relax and Recover 1.13.0 / $Date$
Calculating backup archive size
Backup archive size is 460M (compressed)
Comparing disks.
Disk configuration is identical, proceeding with restore.
Start system layout restoration.
Creating partitions for disk /dev/sda (gpt)
Creating LVM PV /dev/sda3
  0 logical volume(s) in volume group "vg_fedora" now active
Restoring LVM VG vg_fedora
Creating ext4-filesystem / on /dev/mapper/vg_fedora-lv_root
Mounting filesystem /
Creating ext4-filesystem /boot on /dev/sda2
Mounting filesystem /boot
Creating swap on /dev/mapper/vg_fedora-lv_swap
Disk layout created.
Restoring from 'nfs://hpx189.ncsbe.eu.jnj.com/test/fedora/backup.tar.gz'...
Restored 2078 MiB [avg 7440 KiB/sec]OK
Restored 2078 MiB in 287 seconds [avg 7414 KiB/sec]
Installing GRUB2 boot loader
Installation finished. No error reported.

Finished recovering your system. You can explore it under '/mnt/local'.

Finished in 313 seconds
RESCUE fedora:~ # _
```

Recover with rear (4)

```
  Metadata Sequence No  4
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                2
  Open LV               1
  Max PV                0
  Cur PV                1
  Act PV                1
  VG Size               4.50 GiB
  PE Size               32.00 MiB
  Total PE              144
  Alloc PE / Size       144 / 4.50 GiB
  Free  PE / Size       0 / 0
  VG UUID               H7VT2i-mvUY-Y2e5-5adL-bzCw-28CE-gb3Ylx

RESCUE fedora:~ # df
Filesystem                     1K-blocks    Used Available Use% Mounted on
devtmpfs                          435664       0    435664   0% /dev
tmpfs                             456244       0    456244   0% /dev/shm
tmpfs                             456244     200    456044   1% /run
tmpfs                             456244       0    456244   0% /sys/fs/cgroup
/dev/mapper/vg_fedora-lv_root    3128548 2473820    527592  83% /mnt/local
/dev/sda2                         495844   72805    397439  16% /mnt/local/boot
RESCUE fedora:~ # _
```

- Ready? **Reboot (shutdown -r 0)**
- That's it – wait a while for the selinux relabeling
- Verify the restored system

Cloning with rear (1)

- Start the recover process: `rear -v recover`

```
RESCUE beefy:~ # rear -v recover
Relax and Recover 1.13.0 / $Date$
Comparing disks.
Device sda has size 5368709120, 6442450944 expected
Switching to manual disk layout configuration.
Original disk /dev/sda does not exist in the target system. Please choose an app
ropriate replacement.
1) /dev/sda
2) /dev/sdb
3) Do not map disk.
#? 1
```

```
2012-05-15 12:55:03 Disk /dev/sda chosen as replacement for /dev/sda.
Disk /dev/sda chosen as replacement for /dev/sda.
This is the disk mapping table:
    /dev/sda /dev/sda
Please confirm that '/var/lib/rear/layout/disklayout.conf' is as you expect.

1) View disk layout (disklayout.conf)  4) Go to Rear shell
2) Edit disk layout (disklayout.conf)  5) Continue recovery
3) View original disk space usage      6) Abort Rear
#? 1
```

Cloning with rear (2)

```
disk /dev/sda 6442450944 gpt
part /dev/sda 1048576 1048576 rear-noname bios_grub /dev/sda1
part /dev/sda 524288000 2097152 ext4 boot /dev/sda2
part /dev/sda 4843372544 526385152 rear-noname lvm /dev/sda3
# disk /dev/sdb 4294967296 gpt
lvmdev /dev/vg /dev/sda3 WIV8Xr-hN1o-JNRn-XMUU-K16I-I0tF-ErxYUV 11552768
lvmgrp /dev/vg 32768 176 5767168
lvmvol /dev/vg lv_swap 50 3276800
lvmvol /dev/vg lv_root 126 8257536
fs /dev/mapper/vg-lv_root / ext4 uuid=53faa99e-be97-4a15-80d9-936a0103e33e label
= blocksize=4096 reserved_blocks=4% max_mounts=-1 check_interval=0d options=rw,r
elatime,seclabel,user_xattr,barrier=1,data=ordered
fs /dev/sda2 /boot ext4 uuid=576e6373-50c9-4762-8bbd-95f83931a680 label= blocksi
ze=1024 reserved_blocks=5% max_mounts=-1 check_interval=0d options=rw,relatime,s
eclabel,user_xattr,barrier=1,data=ordered
swap /dev/mapper/vg-lv_swap uuid=bf30769d-f25b-4dfd-bd2a-cecf4694e02a label=
/var/lib/rear/layout/disklayout.conf (END)
```

```
1) View disk layout (disklayout.conf)
2) Edit disk layout (disklayout.conf)
3) View original disk space usage
4) Go to Rear shell
5) Continue recovery
6) Abort Rear
#? 5
```

```
Partition rear-noname on /dev/sda: size reduced to fit on disk.
Please confirm that '/var/lib/rear/layout/diskrestore.sh' is as you expect.

1) View restore script (diskrestore.sh)
2) Edit restore script (diskrestore.sh)
3) View original disk space usage
4) Go to Rear shell
5) Continue recovery
6) Abort Rear
#? 1
```

Cloning with rear (3)

```
#!/bin/bash

LogPrint "Start system layout restoration."

mkdir -p /mnt/local
if create_component "vgchange" "rear" ; then
    lvm vgchange -a n >&8
    component_created "vgchange" "rear"
fi

set -e
set -x

if create_component "/dev/sda" "disk" ; then
# Create /dev/sda (disk)
Log "Erasing MBR of disk /dev/sda"
dd if=/dev/zero of=/dev/sda bs=512 count=1
sync
LogPrint "Creating partitions for disk /dev/sda (gpt)"
parted -s /dev/sda mklabel gpt >&2
parted -s /dev/sda mkpart rear-noname 32768B 1081343B >&2
parted -s /dev/sda set 1 bios_grub on >&2
parted -s /dev/sda mkpart ext4 1085440B 525373439B >&2
parted -s /dev/sda set 2 boot on >&2
/var/lib/rear/layout/diskrestore.sh
```

```
1) View restore script (diskrestore.sh)
2) Edit restore script (diskrestore.sh)
3) View original disk space usage
4) Go to Rear shell
5) Continue recovery
6) Abort Rear
#? 5
```

Cloning with rear (4)

```
Start system layout restoration.
Creating partitions for disk /dev/sda (gpt)
Creating LVM PV /dev/sda3
  0 logical volume(s) in volume group "vg" now active
Creating LVM VG vg
Creating LVM volume vg/lv_swap
Creating LVM volume vg/lv_root
An error occured during layout recreation.

1) View Rear log
2) View original disk space usage
3) Go to Rear shell
4) Edit restore script (diskrestore.sh)
5) Continue restore script
6) Abort Rear
#? 1
```

```
+++ echo -e 'Creating LVM volume vg/lv_root'
+++ lvm lvcreate -l 126 -n lv_root vg
  Volume group "vg" has insufficient free space (94 extents): 126 required.
2012-05-15 13:08:28 An error occured during layout recreation.
```

```
1) View Rear log
2) View original disk space usage
3) Go to Rear shell
4) Edit restore script (diskrestore.sh)
5) Continue restore script
6) Abort Rear
#? 4
```

```
LogPrint "Creating LVM volume vg/lv_root"
lvm lvcreate -l 126 -n lv_root vg >&2
```

```
lvm lvcreate -l 94 -n lv_root vg >&2
```

Cloning with rear (5)

```
1) View Rear log
2) View original disk space usage
3) Go to Rear shell
4) Edit restore script (diskrestore.sh)
5) Continue restore script
6) Abort Rear
#? 5
```

```
#? 5
Start system layout restoration.
Skipping /dev/sda (disk) as it has already been created.
Skipping /dev/sda1 (part) as it has already been created.
Skipping /dev/sda2 (part) as it has already been created.
Skipping /dev/sda3 (part) as it has already been created.
Skipping pv:/dev/sda3 (lvmdev) as it has already been created.
Skipping /dev/vg (lvmgrp) as it has already been created.
Skipping /dev/mapper/vg-lv_swap (lvmvol) as it has already been created.
Creating LVM volume vg/lv_root
Creating ext4-filesystem / on /dev/mapper/vg-lv_root
Mounting filesystem /
Creating ext4-filesystem /boot on /dev/sda2
Mounting filesystem /boot
Creating swap on /dev/mapper/vg-lv_swap
Disk layout created.
Please start the restore process on your backup host.

Make sure that you restore the data into '/mnt/local' instead of '/' because the
hard disks of the recovered system are mounted there.

Please restore your backup in the provided shell and, when finished, type exit
in the shell to continue recovery.
rear>
```

Cloning with rear (6)

```
RESCUE beefy:~ # cat /etc/rear/local.conf
# sample local configuration

# Create Rear rescue media as ISO image
OUTPUT=ISO

# optionally define (non-default) backup software, e.g. TSM, NBU, DP, BACULA
# BACKUP=TSM

# the following is required on older VMware VMs
MODULES_LOAD=( vmxnet )

# to see boot messages on the serial console (uncomment next line)
# KERNEL_CMDLINE="console=tty0 console=ttyS1"
RESCUE beefy:~ # df
Filesystem            1K-blocks   Used Available Use% Mounted on
devtmpfs                 425332      0    425332   0% /dev
tmpfs                    453128      0    453128   0% /dev/shm
tmpfs                    453128    204    452924   1% /run
tmpfs                    453128      0    453128   0% /sys/fs/cgroup
/dev/mapper/vg-lv_root  3072888 111288   2838396   4% /mnt/local
/dev/sda2                508745  23411    459734   5% /mnt/local/boot
RESCUE beefy:~ #
```

So you better know what you're doing, right?
The **BACKUP** variable was not set in the **/etc/rear/local.conf** configuration file!

Get your hands dirty?

- We hope you want to dig deeper into rear!
- Getting started:
  - Use: rear -s mkbackup
    to see the flow of the scripts it will execute
  - Depends on BACKUP method, architecture and OS version/brand
  - Be careful: rear -s recover
    follows a different flow (seems logically, but you must understand the difference)

Where is the code?

- Main script is /usr/sbin/rear
- All the other scripts live under /usr/share/rear
- Documentation is at /usr/share/doc/rear-*x.y.z*
- ***Good news! It's all written in Bash***

Where to put a script?

- **mkbackup** method: /usr/share/rear/...
  - conf/ - configuration files (/etc/rear/*.conf read last)
  - prep/ - preparation work; checking the environment
  - layout/save/ - save the disk layout /var/lib/rear/layout
  - rescue/ - modules, network, storage,...
  - build/ - populate the initial ramdisk for our rescue image
  - pack/ - create the initrd and copy kernel
  - output/ - create the ISO image and copy to OUTPUT_URL
  - backup/ - make the backup archive to BACKUP_URL

Where to put a script? (2)

- **recover** method: /usr/share/rear/...
  - conf/ - read the configuration file + /etc/rear/*.conf
  - setup/ - user defined scripts to run before recover
  - verify/ - to check if a recover is possible at all
  - layout/prepare – recreate the disk layout
  - restore/ - restore the archive from BACKUP_URL
  - finalize/ - do some dirty tricks for disks, grub,...
  - wrapup/ - copy the recover log to /mnt/local/root/

Example script: sysreqs.sh

- A simple script to save basic system requirements – sysreqs.sh
  - OS version; rear version
  - CPU, memory
  - Disk space requirements
  - IP addresses in use; routes
- Copy sysreqs.sh to a flow, e.g. rescue is a good choice
  - # cp /tmp/sysreqs.sh \ /usr/share/rear/rescue/GNU/Linux/96_sysreqs.sh

Test the script

- # rear -s mkrescue | grep sysreqs
  Source rescue/GNU/Linux/96_sysreqs.sh
- # rear -v mkrescue
- # cat /var/lib/rear/sysreqs/Minimal_System_Requirements.txt

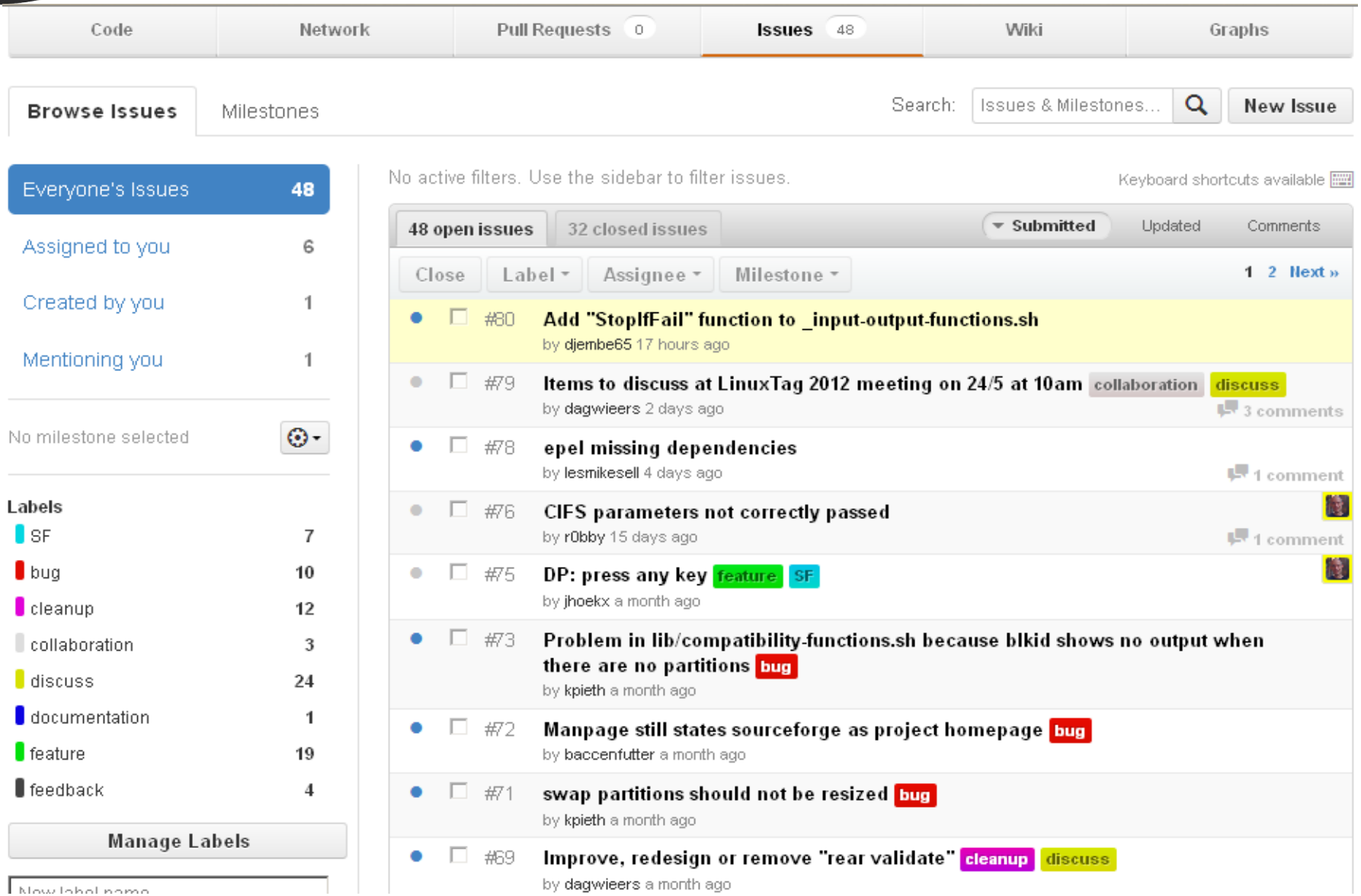
```
fedora - 2012-05-22 11:26
Operating system:
LSB Version:    :core-4.0-ia32:core-4.0-noarch
Distributor ID: Fedora
Description:    Fedora release 16 (Verne)
Release:        16
Codename:       Verne
Relax and recover version:
Relax and Recover 1.13.0 / $Date$

There are 1 CPU core(s) at  2393.832 MHz
748 MiB of physical memory

Disk space requirements:
  OS (vg + swap + /boot)
    size: 6.09 GiB

Network Information:
  IP adresses:
    ip 6 ::1 subnet /128 scope host DNS name
    ip 192.168.5.135 subnet /24 DNS name
    ip 6 fe80::20c:29ff:fe63:5cd0 subnet /64 scope link DNS name
```

https://github.com/rear/rear/issues



Contacts

Web-site: http://rear.github.com/
GitHib: https://github.com/rear/rear
Mailing list: rear-users@lists.sourceforge.net

Rear Maintainer - **Gratien D'haese** - gratien.dhaese@it3.be

Rear Maintainer - **Schlomo Schapiro** - schlomo@schapiro.org

Rear Developer – **Jeroen Hoekx -** jeroen.hoekx@hamok.be

Rear Developer **– Dag Wieers -** dag@wieers.com