
OPEN SOURCE BACKUP CONFERENCE

# Relax-and-Recover (ReaR) Automated Testing with Bareos

Gratien D'haese
IT3 Consultants

https://gdha.github.io/rear-automated-testing/


# Agenda

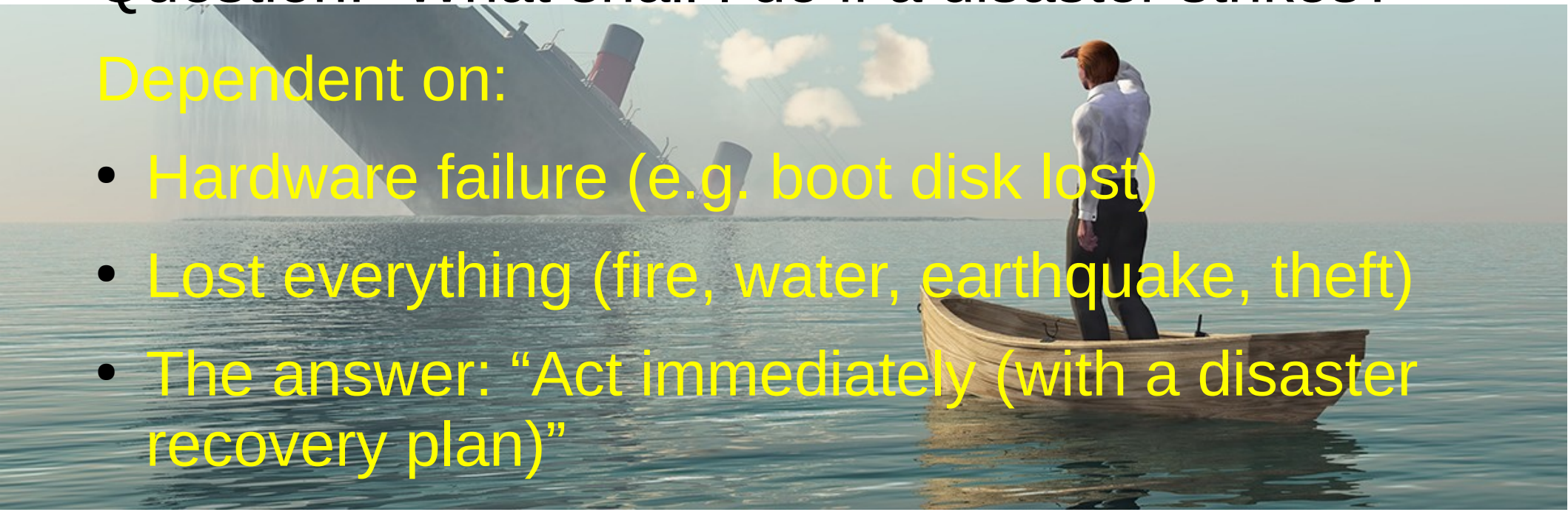
- Who am I?
- What is Relax-and-Recover?
- Relax-and-Recover Automated Testing with Bareos
- Live DEMO

# Who am I?

- Gratien D'haese
- IT3 Consultants (company)
  - > 30 years Unix experience
  - Unix/Linux Engineer (incl. DevOps)
  - ReaR Support Contracts
- Relax-and-Recover (ReaR)
  - Major Open Source project
  - https://github.com/gdha

# Linux Disaster Recovery

Question: “What shall I do if a disaster strikes?”

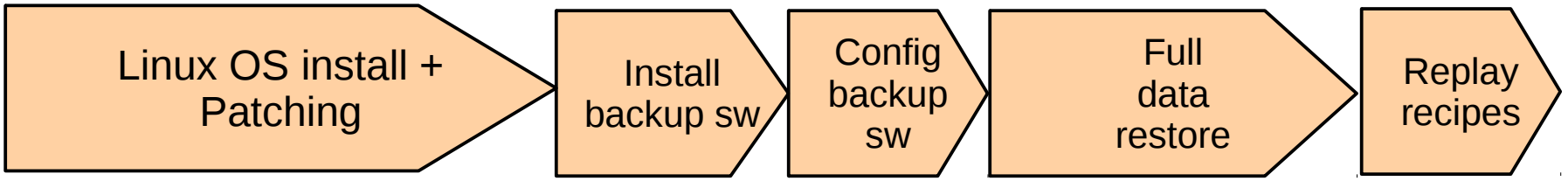


Dependent on:

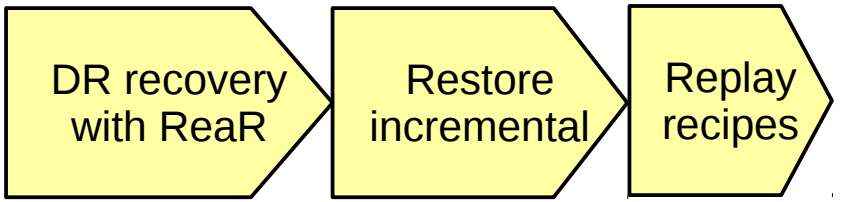
- Hardware failure (e.g. boot disk lost)
- Lost everything (fire, water, earthquake, theft)
- The answer: “Act immediately (with a disaster recovery plan)”
- Use a good backup solution such as **Bareos**
- Use a good DR solution such as **ReaR**

# Why are backups not enough?

Disaster Recovery via re-install of Linux OS and restore data




Disaster Recovery with ReaR and restore data





# Relax-and-Recover (ReaR) as DR solution

- ReaR is a tool that implements a DR work-flow
- Basically meaning:
  - Modular framework written in **Bash**
  - Easy to extend to own needs
  - Easy to deploy (set up and forget)
  - Integration for various Linux technologies
  - Integration with various back-up solutions
  - Attempts to make system recovery *as easy as possible*


# ReaR Maintainers (alphabetical order)

- Sébastien Chabrolles (France)
  https://github.com/schabrolles
- Gratien D'haese (Belgium)
  https://github.com/gdha
- Vladimir Gozora (Slovakia)
  https://github.com/gozora
- Johannes Meixner (Germany)
  https://github.com/jsmeix
- Schlomo Schapiro (Germany)
  https://github.com/schlomo
- And many more contributors

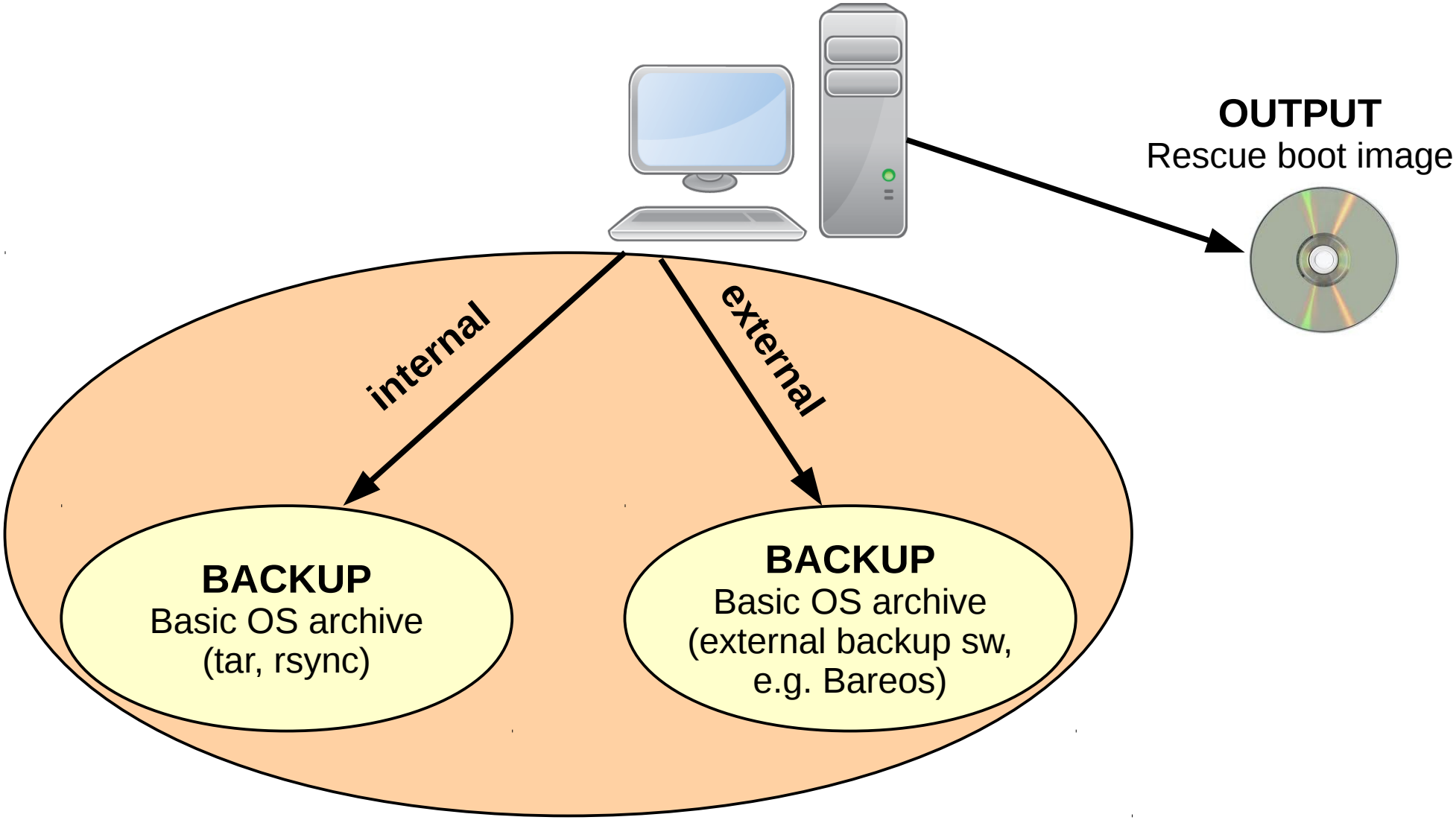

# Disaster Recovery – How It Works

- Store the disk layout
  - Partitioning, LVM and RAID configuration
  - File systems, file system labels ...
  - Boot loader (GRUB, GRUB2, LILO, UEFI)
- Store the files (tar, rsync, through backup software such as Bareos)
- Create bootable rescue media with system configuration (and backup data)
- **Can be done online** (no business interruption)

# Disaster Recovery – Rescue Media

- Create “rescue linux” from running system
- Optimally compatible “tool box”
- Clone the system environment
  - Linux kernel and modules
  - Device driver configuration
  - Network configuration
  - Basic system software and tools
- Operate entirely in RAM (initrd)

# DR Flow – BACKUP and OUTPUT




# Usage of rear

- Shell scripts are stored under /usr/share/rear
- Scripts are executed via work-flows:
  - mkrescue (only make rescue image)
  - mkbackup (including make rescue image)
  - mkbackuponly (excluding make rescue image)
  - recover (the actual recovery part)
- Easy to incorporate new scripts, e.g. for information gathering of Hard- and Software, or other goodies


# Getting started with ReaR

- Download it from
  - Our web-site
    - http://relax-and-recover.org/download/
  - The rear-snapshot rpm's build from GitHub
    - http://download.opensuse.org/repositories/Archiving:/Backup:/Rear:/Snapshot/
  - The official source
    - https://github.com/rear/rear
  - The official repo's (Fedora, RHEL and SLES)
    - yum install rear
    - zypper install rear

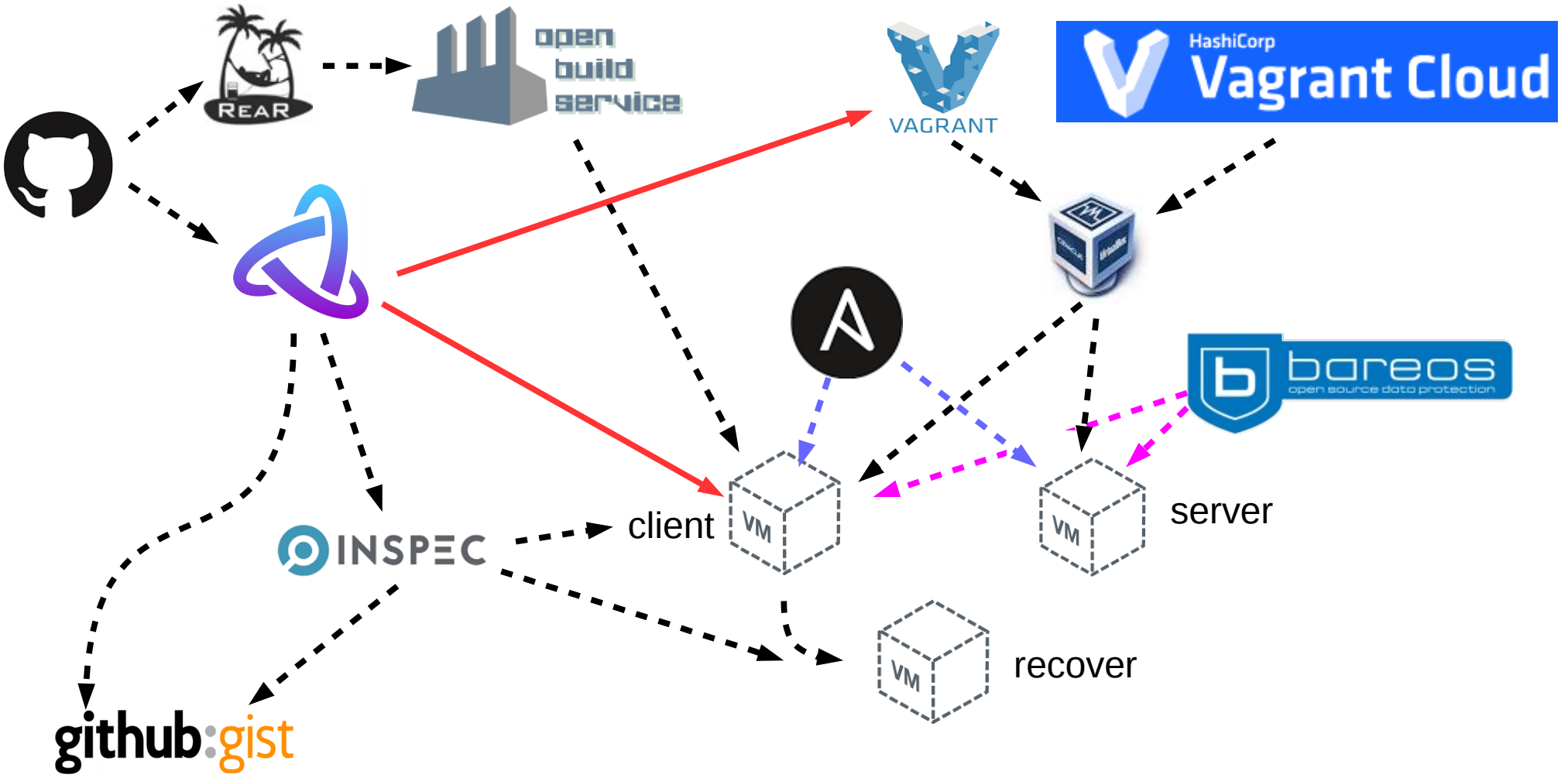

# Testing ReaR

- ReaR is due to a wide range of options difficult to test
- Different Linux flavors are using similar tools with some minor differences :-(
- Too much to be able to test everything before a new release, and sometimes it is even not possible just because we do not have the hardware
- Is and will stay a challenge for the future...


# ReaR Builds

- Finding a right balance between CI Testing and Automated ReaR Testing
- Via OpenSuse Build Services we build daily a fresh ReaR snapshot package (for free)
- We have some excellent ReaR developers who do lots of coding (for free)
- Support via GitHub issues
  - Free support
  - Commercial support

# ReaR Automated Testing Workflow



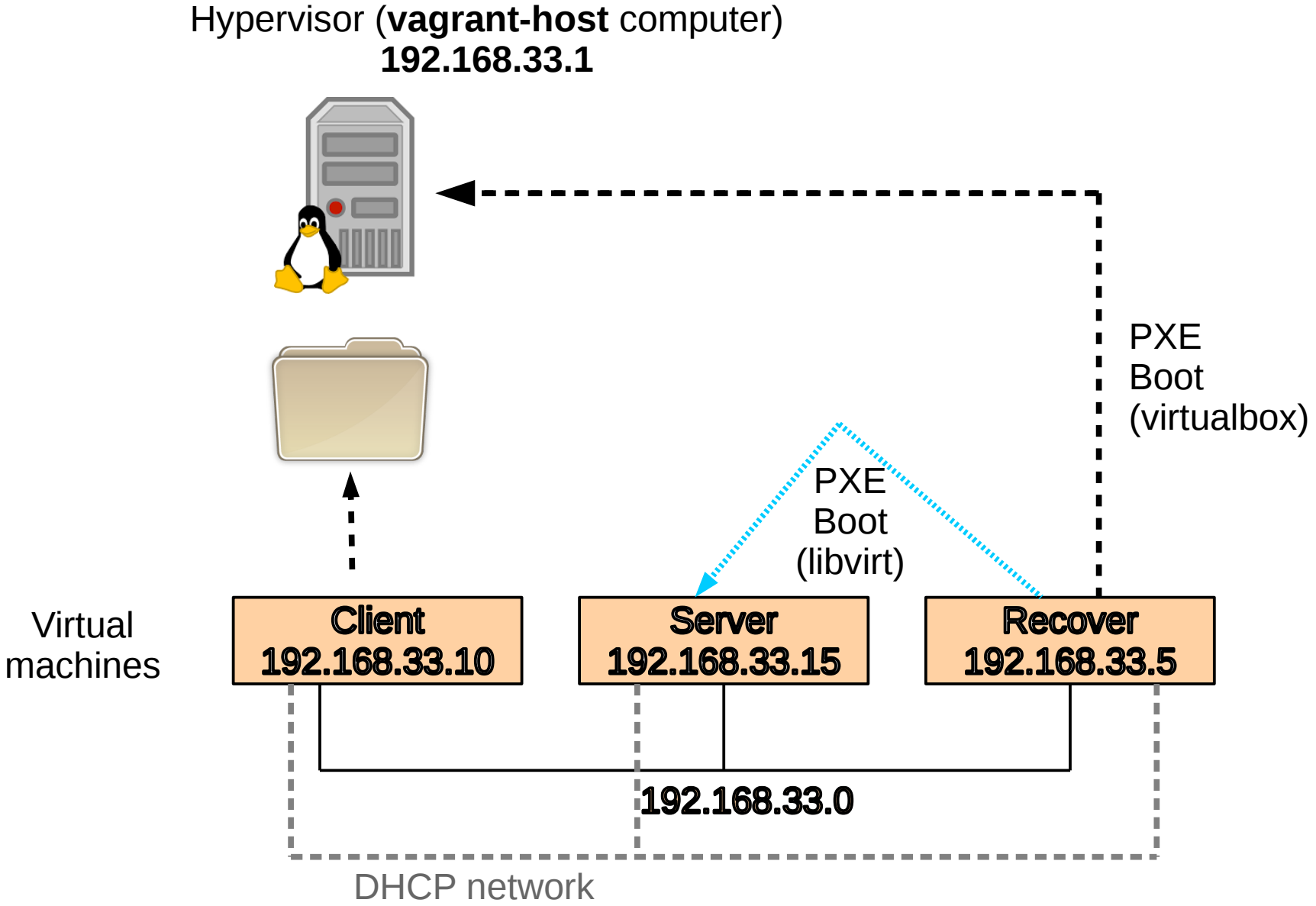

# Automated ReaR Testing

- Wrote it for customers with a ReaR support or subscription contract
- Currently we support the following GNU/Linux distributions:
  - CentOS 7
  - Ubuntu 14.04, Ubuntu 16.04
  - SLES 11, SLES 12
- The VMs are provisioned with ansible playbooks


# Automated ReaR Testing (cont'd)

- We start the process and it automatically does
  - DR image creation
  - Backup of system via one of the following methods:
    - BACKUP=NETFS (with GNU tar)
    - BACKUP=DUPLICITY
    - BACKUP=BAREOS
  - PXE or ISO boot the recover system with DR image made
  - Restore backup
  - Reboot the recover system

# Test Configuration




# Set up vagrant environment

- Host system must be GNU/Linux, or Mac OS based
- A hypervisor like VirtualBox (or KVM on Linux)
- Install “vagrant” from your distro, or from https://www.vagrantup.com/downloads.html
- KVM with libvirt needs the vagrant-libvirt plugin vagrant plugin install vagrant-libvirt
- Install “git” software to clone the Vagrantfile and scripts


# Install the ReaR automated Testing software

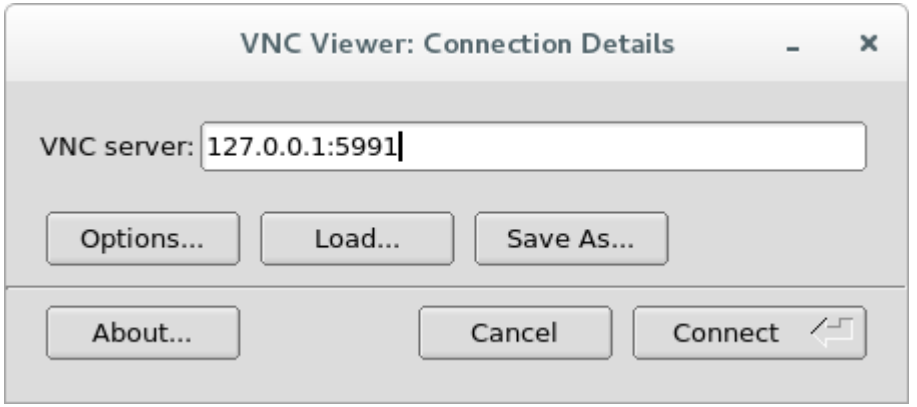
- Is Open Source and licensed under GPLv3
- New code is written only for customers with a valid ReaR Support contract *(PR are welcome)*
- git clone git@github.com:gdha/rear-automated-testing.git
- Go into directory “rear-automated-testing”
- Type “./rear-automated-test.sh -h” to see info
- Uses “*vagrant*” to drive the creation of the VMs
- Account vagrant/vagrant (and root/vagrant)


# Login via vagrant or ssh

- Once the VMs are up and running
- Login via vagrant:
  - sudo vagrant ssh client
  - sudo vagrant ssh server
- Another way to login is via ssh:
  - ssh root@192.168.33.10  (client root pw is vagrant)
  - ssh root@192.168.33.15  (server root pw is vagrant)

# Login via vncviewer

- If you install “tigervnc” you can use vncviewer
- Use address 127.0.0.1 (localhost)
- Port 5991 for “client”
- Port 5992 for “server”
- Port 5993 for “recover”



# Try it yourself?

- https://github.com/gdha/rear-automated-testing

```
 $ sudo ./rear-automated-test.sh -b ISO -s 2.4 -c
templates/BAREOS-with-ISO.conf
+-----------------------------------------------+
|    Relax-and-Recover Automated Testing script |
|              version 1.4                      |
+-----------------------------------------------+
Command line options: rear-automated-test.sh -b ISO -s
2.4 -c templates/BAREOS-with-ISO.conf
Distribution: centos7
Boot method: ISO
ReaR version: 2.4
Provider: virtualbox
ReaR configuration: BAREOS-with-ISO.conf
Log file: /export/rear-tests/logs/2018-09-17_09-28-
19/rear-automated-test.sh.log
```


# ReaR Automated Testing

- ReaR Automated Testing speeds up
  - Validation processes
  - Bug hunting
  - Compliance checks
  - Trusworthy of “unstable” ReaR repository
  - Acceptance of ReaR within corporations and Open Source vendors
  - Stable releases can be tested anytime
- We accept ***pull requests*** and sponsoring


# /etc/rear/local.conf used

```
OUTPUT=ISO
OUTPUT_URL=nfs://10.0.2.2/root/.config/VirtualBox/TFTP/isos
OUTPUT_OPTIONS="nfsvers=3,nolock"
BACKUP=BAREOS
BAREOS_RESTORE_JOB=client-restore
BAREOS_FILESET=client-fileset
BAREOS_RECOVERY_MODE="automatic"
PRE_BACKUP_SCRIPT=/usr/local/bin/client-backup-with-bareos
PROGS=( "${PROGS[@]}" showmount mount.nfs umount.nfs )
MODULES=( "${MODULES[@]}" nfs )
PRE_RECOVERY_SCRIPT="systemctl start rpcbind.target || rpcbind &"
PXE_CONFIG_URL=nfs://10.0.2.2/root/.config/VirtualBox/TFTP/pxelinux.cfg
ISO_DEFAULT="automatic"
ISO_RECOVER_MODE="unattended"
USE_STATIC_NETWORKING=y
KERNEL_CMDLINE="$KERNEL_CMDLINE net.ifnames=0"
FIRMWARE_FILES=( 'no' )
SSH_ROOT_PASSWORD="vagrant"
TIMESYNC=NTPDATE
TIMESYNC_SOURCE=0.pool.ntp.org
TEST_LOG_DIR_URL=nfs://10.0.2.2/export/rear-tests/logs/2018-08-21_12-50-36
```

# ReaR Automated Testing

demonstration

# Need more of this?

We can foresee in a customized workshop on consultancy basis, or set-up in-house full automated ReaR testing for customers with a valid support contract

See http://www.it3.be/rear-support


**Gratien D'haese**
**IT3 Consultants**


# Backup Slides in case live demonstration is not possible


# The help and usage page

```
$ sudo ./rear-automated-test.sh -h
Usage: rear-automated-test.sh [-d distro] [-b <boot method>] [-s <stable
rear version>] [-p provider] [-c rear-config-file.conf] [-t test] -vh
        -d: The distribution to use for this automated test (default: centos7)
        -b: The boot method to use by our automated test (default: PXE)
        -s: The <stable rear version> is the specific version we want to test,
e.g. 2.3 (default: <empty> )
        -p: The vagrant <provider> to use (default: virtualbox)
        -c: The ReaR config file we want to use with this test (default: PXE-
booting-with-URL-style.conf)
        -l: The ReaR test logs top directory (default: /export/rear-tests/logs)
        -t: The ReaR validation test directory (see tests directory; no default)
        -h: This help message.
        -v: Revision number of this script.
```


# Starting the script

$ sudo ./rear-automated-test.sh -b ISO -s 2.4 -c templates/BAREOS-with-ISO.conf

```
+-----------------------------------------------+
|    Relax-and-Recover Automated Testing script |
|                    version 1.4                |
+-----------------------------------------------+
```

Author: Gratien D'haese
Copyright: GPL v3

Command line options: rear-automated-test.sh -b ISO -s 2.4 -c templates/BAREOS-with-ISO.conf
Distribution: centos7
Boot method: ISO
ReaR version: 2.4
Provider: virtualbox
ReaR configuration: BAREOS-with-ISO.conf
Log file: /export/rear-tests/logs/2018-09-17_09-28-19/rear-automated-test.sh.log


Using virtualbox as hypervisor
Current distro directory is centos7
Copy the Vagrantfile.virtualbox to Vagrantfile
Bringing up the vagrant VMs client and server
Bringing machine 'client' up with 'virtualbox' provider...
Bringing machine 'server' up with 'virtualbox' provider...
==> client: Checking if box 'centos/7' is up to date...
==> client: Machine already provisioned. Run `vagrant provision` or use the `--provision`
==> client: flag to force provisioning. Provisioners marked to run always will still run.
==> server: Checking if box 'centos/7' is up to date...
==> server: Machine already provisioned. Run `vagrant provision` or use the `--provision`
==> server: flag to force provisioning. Provisioners marked to run always will still run.

Sleep for 5 seconds [Control-C is now possible]

Do not use Control-C anymore, or the VMs will be destroyed
---
Current machine states:

client running (virtualbox)
server running (virtualbox)
recover poweroff (virtualbox)


Check if  eth1 is active on client [known issue https://github.com/mitchellh/vagrant/issues/8166]
Check if  eth1 is active on server
Doing ping tests to VMs client and server
client is up and running - ping test OK
server is up and running - ping test OK


## Install stable ReaR version 2.4 on the VM client

Resolving Dependencies
--> Running transaction check
---> Package rear.x86_64 0:2.4-1.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

```
================================================================================
 Package     Arch         Version          Repository                  Size
================================================================================
Installing:
 rear        x86_64       2.4-1.el7        Archiving_Backup_Rear       587 k

Transaction Summary
================================================================================
Install  1 Package
```

## ReaR version that will be tested is:

Relax-and-Recover 2.4 / 2018-06-21

## Content of /etc/rear/local.conf is:

```
OUTPUT=ISO
OUTPUT_URL=nfs://10.0.2.2/root/.config/VirtualBox/TFTP/isos
OUTPUT_OPTIONS="nfsvers=3,nolock"
BACKUP=BAREOS
BAREOS_RESTORE_JOB=client-restore
BAREOS_FILESET=client-fileset
BAREOS_RECOVERY_MODE="automatic"
PRE_BACKUP_SCRIPT=/usr/local/bin/client-backup-with-bareos
PROGS=( "${PROGS[@]}" showmount mount.nfs umount.nfs )
MODULES=( "${MODULES[@]}" nfs )
PRE_RECOVERY_SCRIPT="systemctl start rpcbind.target || rpcbind &"
PXE_CONFIG_URL=nfs://10.0.2.2/root/.config/VirtualBox/TFTP/pxelinux.cfg
ISO_DEFAULT="automatic"
ISO_RECOVER_MODE="unattended"
USE_STATIC_NETWORKING=y
KERNEL_CMDLINE="$KERNEL_CMDLINE net.ifnames=0"
FIRMWARE_FILES=( 'no' )
SSH_ROOT_PASSWORD="vagrant"
TIMESYNC=NTPDATE
TIMESYNC_SOURCE=0.pool.ntp.org
TEST_LOG_DIR_URL=nfs://10.0.2.2/export/rear-tests/logs/2018-09-17_09-28-19
```

## Run rear -v mkbackup

Relax-and-Recover 2.4 / 2018-06-21
Using log file: /var/log/rear/rear-client.log
Creating disk layout
Using guessed bootloader 'GRUB' (found in first bytes on /dev/sda)
Creating root filesystem layout
Adding biosdevname=0 to KERNEL_CMDLINE
Copying logfile /var/log/rear/rear-client.log into initramfs as '/tmp/rear-client-partial-2018-09-17T09:29:10+0200.log'
Copying files and directories
Copying binaries and libraries
Copying kernel modules
Omit copying files in /lib*/firmware/ (FIRMWARE_FILES='no')
Creating recovery/rescue system initramfs/initrd initrd.cgz with gzip default compression
Created initrd.cgz with gzip default compression (56761531 bytes) in 7 seconds
Making ISO image
Wrote ISO image: /var/lib/rear/output/rear-client.iso (63M)
Copying resulting files to nfs location
Saving /var/log/rear/rear-client.log as rear-client.log to nfs location
Save the /var/log/rear/rear-client.log to nfs://10.0.2.2/export/rear-tests/logs/2018-09-17_09-28-19
Exiting rear mkbackup (PID 4197) and its descendant processes
Running exit tasks

The rear mkbackup was successful


Copy PXE configuration entry to pxelinux.cfg to enable ISO boot menu entry

Profile: **InSpec** Profile (compliance-checks)
Version: 0.1.0
Target:  ssh://root@client:22

- ✔ kernel.shmall: kernel.shmall check
  - ✔ Kernel Parameter kernel.shmall value should eq 2097152
- ✔ kernel.shmmax: kernel.shmmax check
  - ✔ Kernel Parameter kernel.shmmax value should eq 134217728
- ✔ fs.file-max: fs.file-max check
  - ✔ Kernel Parameter fs.file-max value should eq 65536
- ✔ filesystem-root: Verify / directory
  - ✔ File / should be directory
- ✔ home-vagrant-exists: Verify /home/vagrant directory
  - ✔ File /home/vagrant should be directory
- ✔ iputils integrity: RPM integrity test on iputils package
  - ✔ System Package iputils should be installed
  - ✔ Command rpm -V iputils stdout should eq ""
- ✔ root-account: The super user account
  - ✔ User root should exist

Profile Summary: 10 successful controls, 0 control failures, 0 controls skipped
Test Summary: 24 successful, 0 failures, 0 skipped


**Halting the client VM before doing the recovery**

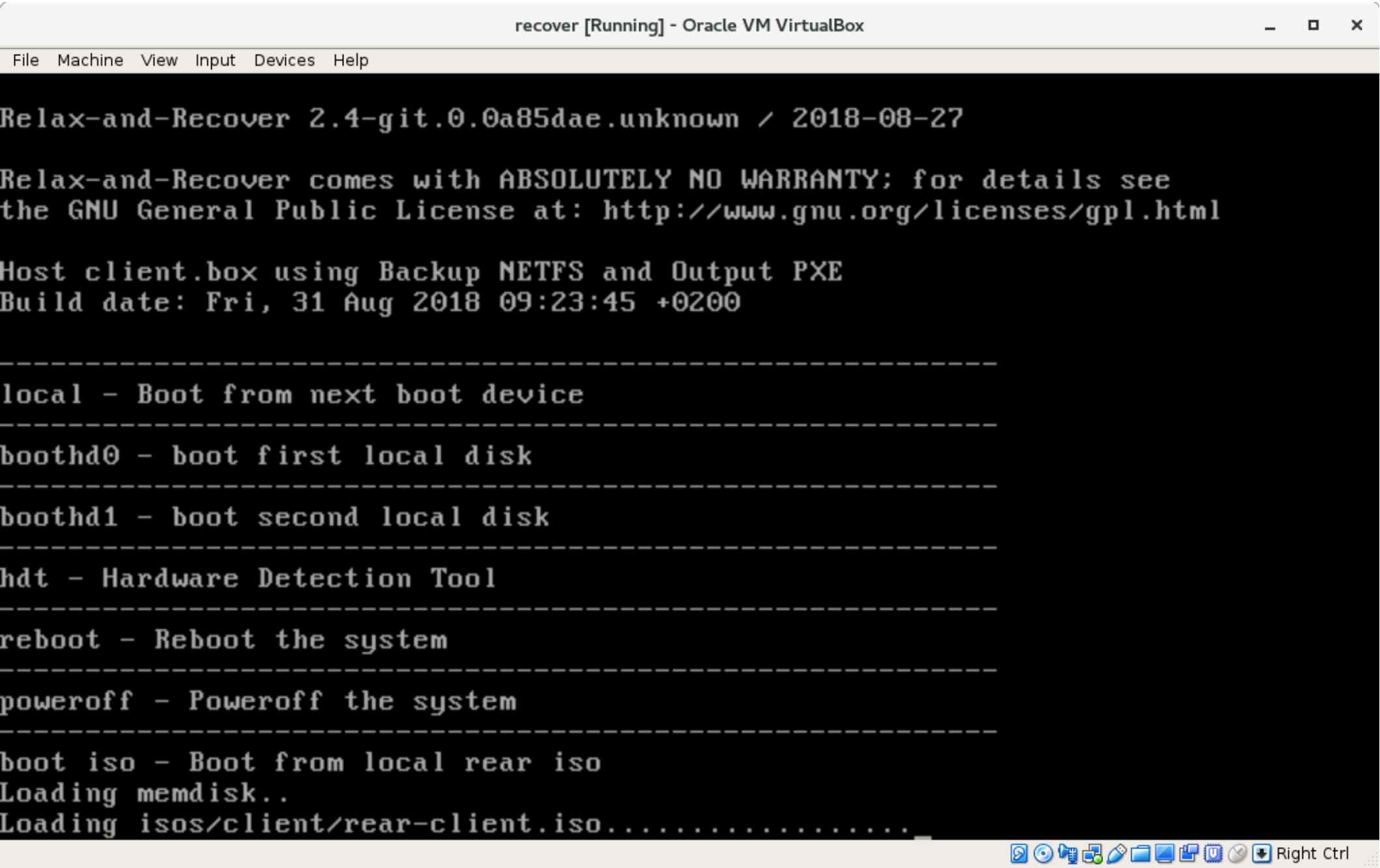
Recover VM will use the client IP address after it has been fully restored

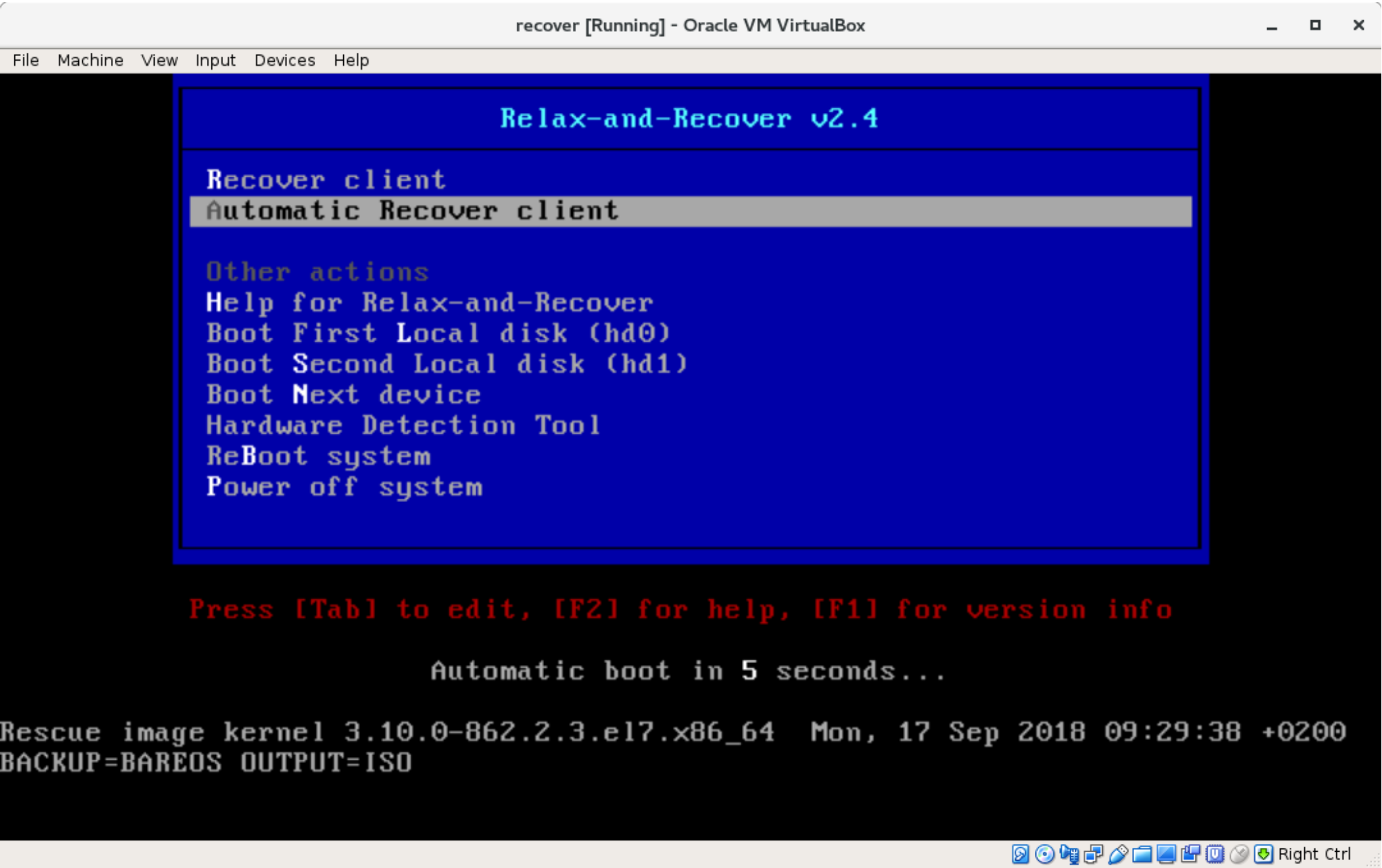
==> client: Attempting graceful shutdown of VM...

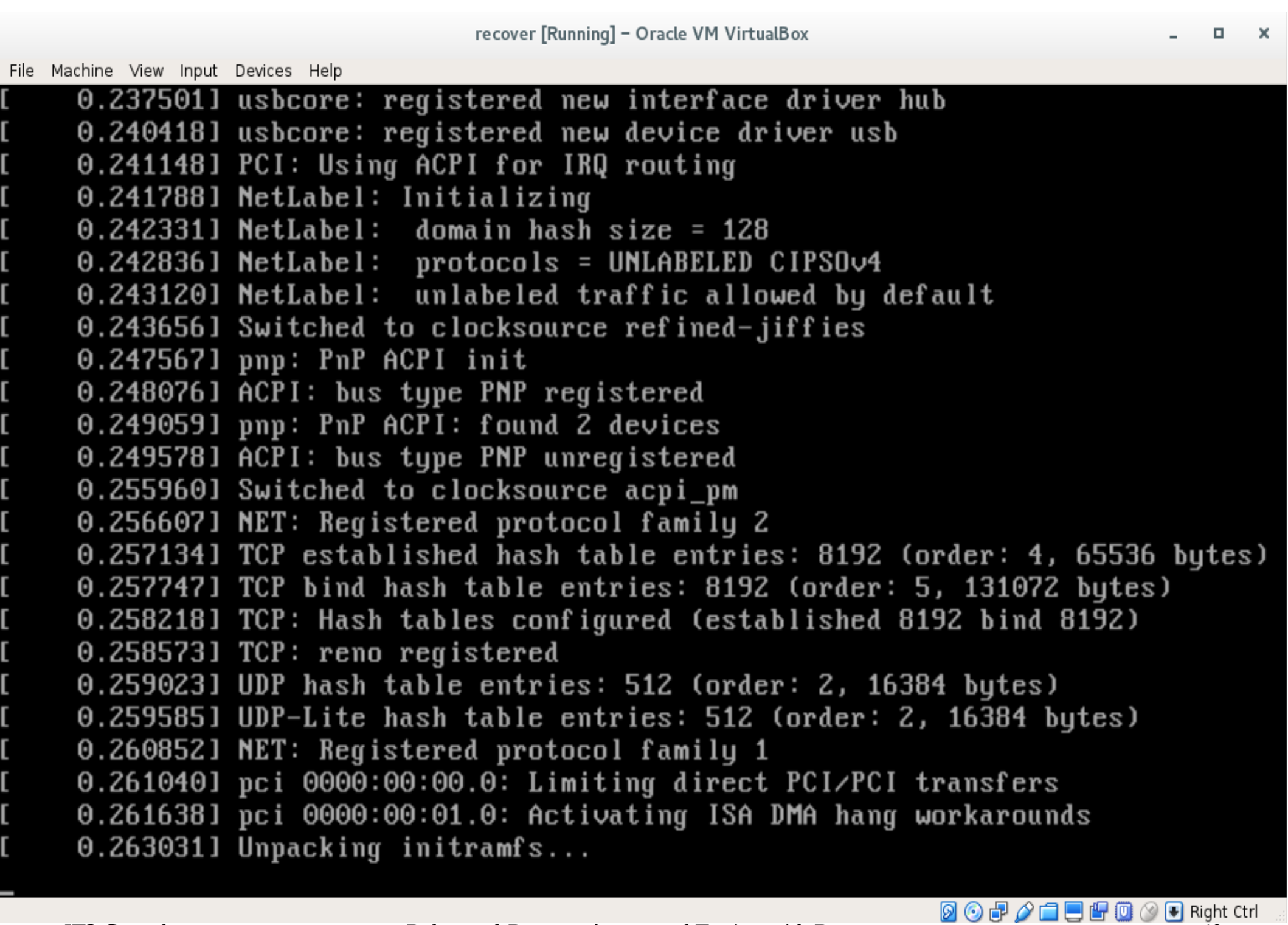
Copied private key of client VB to recover VB config area

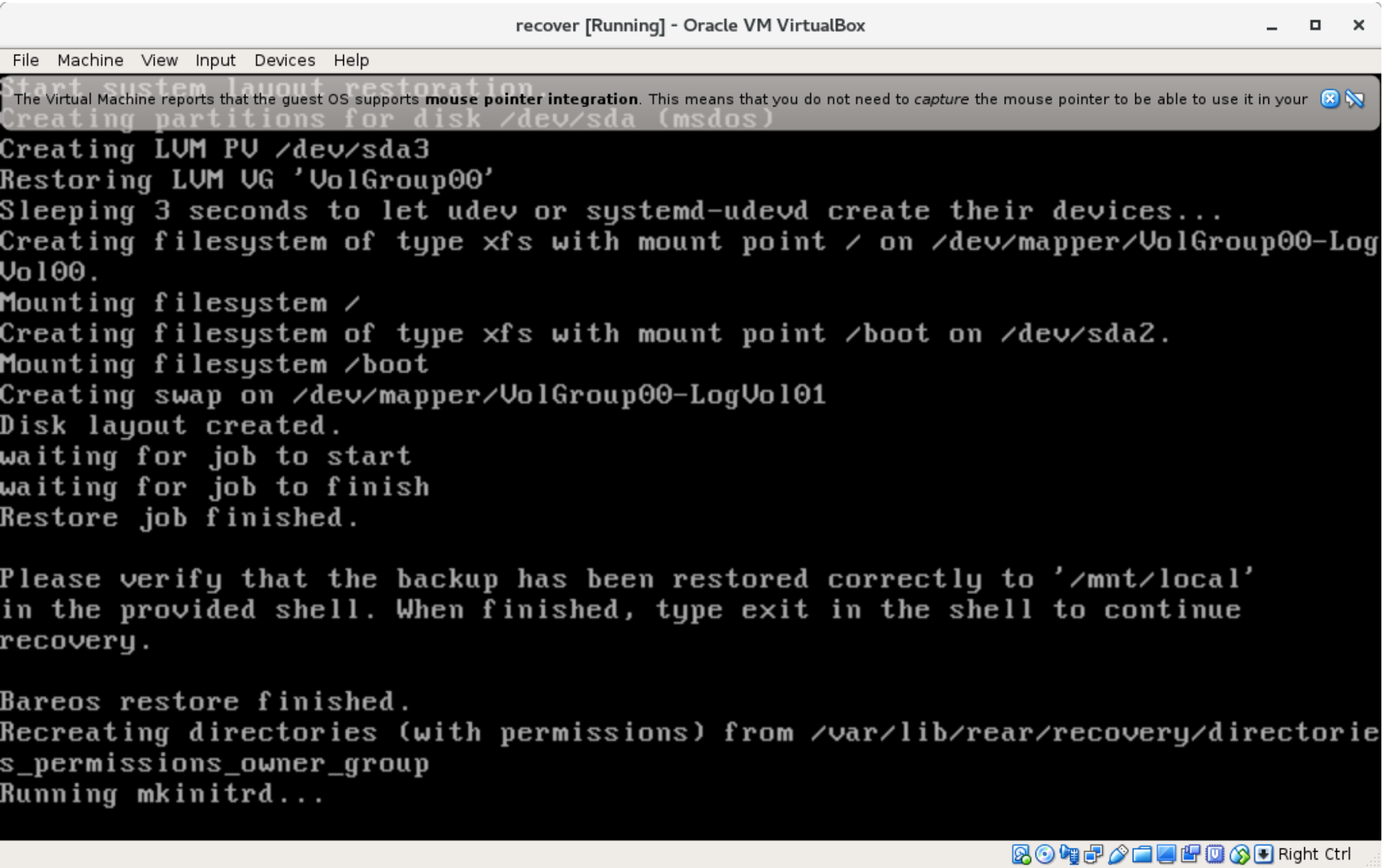
**Starting the recover VM**

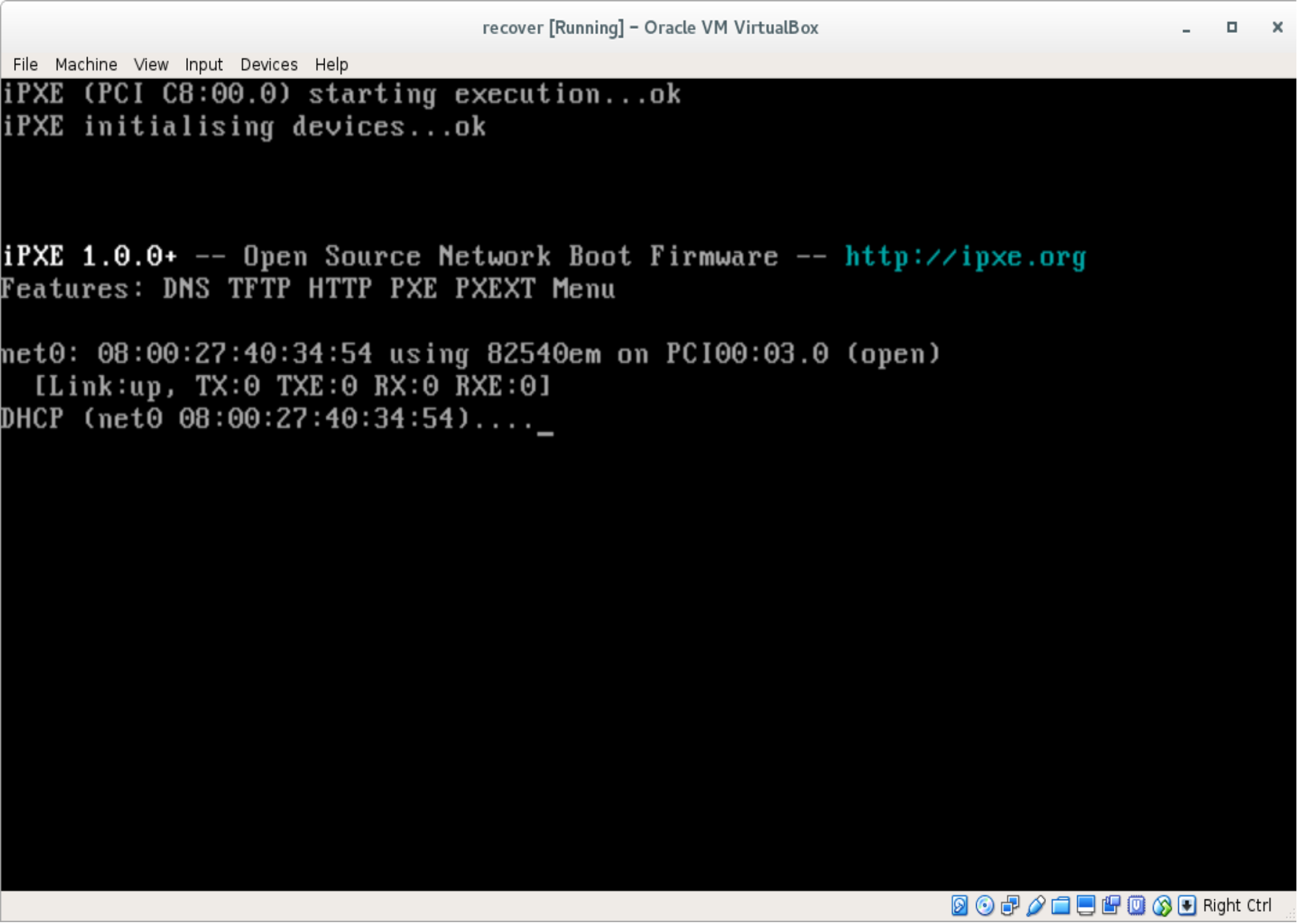
Bringing machine 'recover' up with 'virtualbox' provider...
==> recover: Checking if box 'clink15/pxe' is up to date...

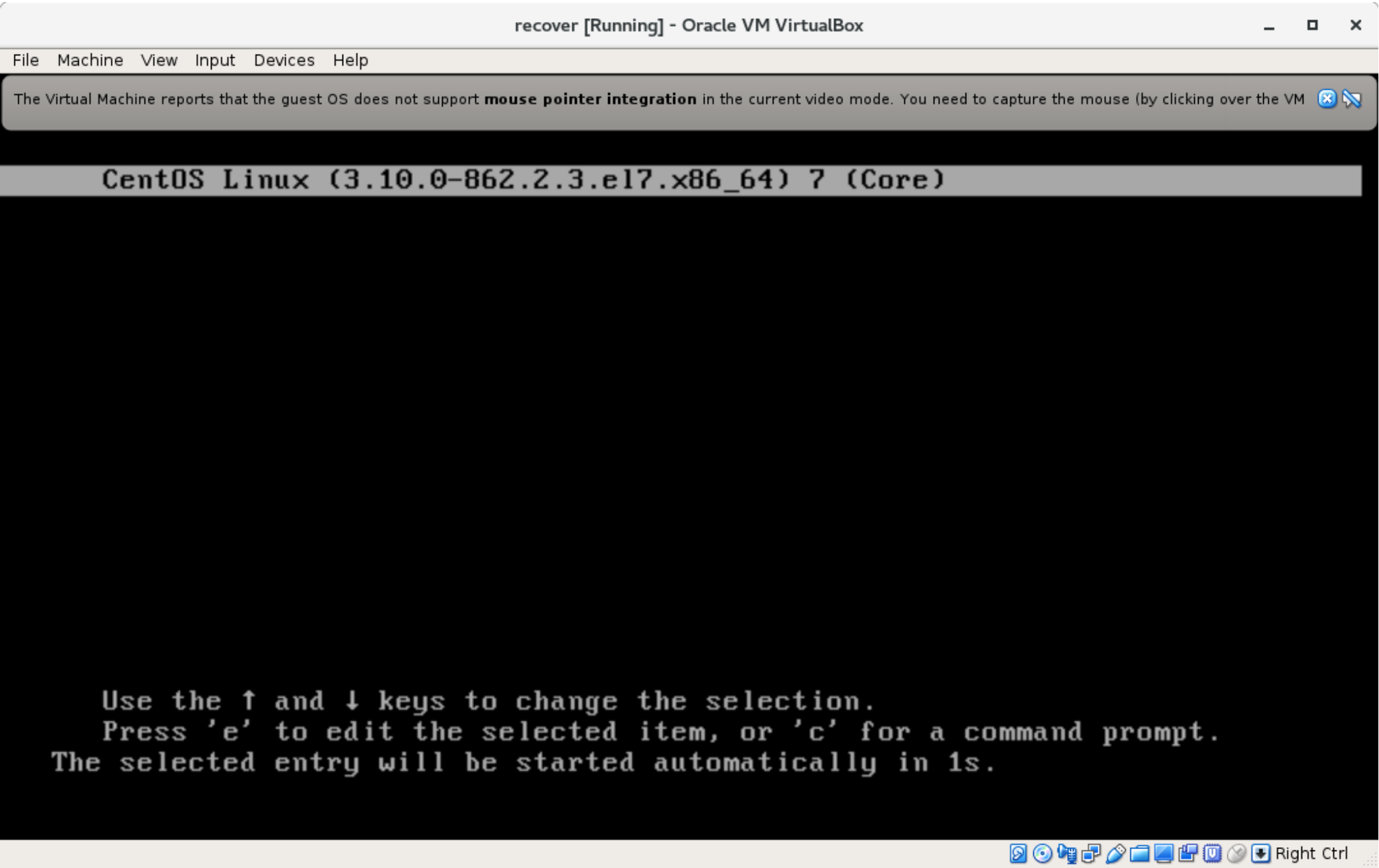

recover [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Relax-and-Recover 2.4-git.0.0a85dae.unknown / 2018-08-27
Relax-and-Recover comes with ABSOLUTELY NO WARRANTY; for details see
the GNU General Public License at: http://www.gnu.org/licenses/gpl.html
Host client.box using Backup NETFS and Output PXE
Build date: Fri, 31 Aug 2018 09:23:45 +0200
local - Boot from next boot device
boothd0 - boot first local disk
boothd1 - boot second local disk
hdt - Hardware Detection Tool
reboot - Reboot the system
poweroff - Poweroff the system
boot iso - Boot from local rear iso
Loading memdisk..
Loading isos/client/rear-client.iso.....
Right Ctrl

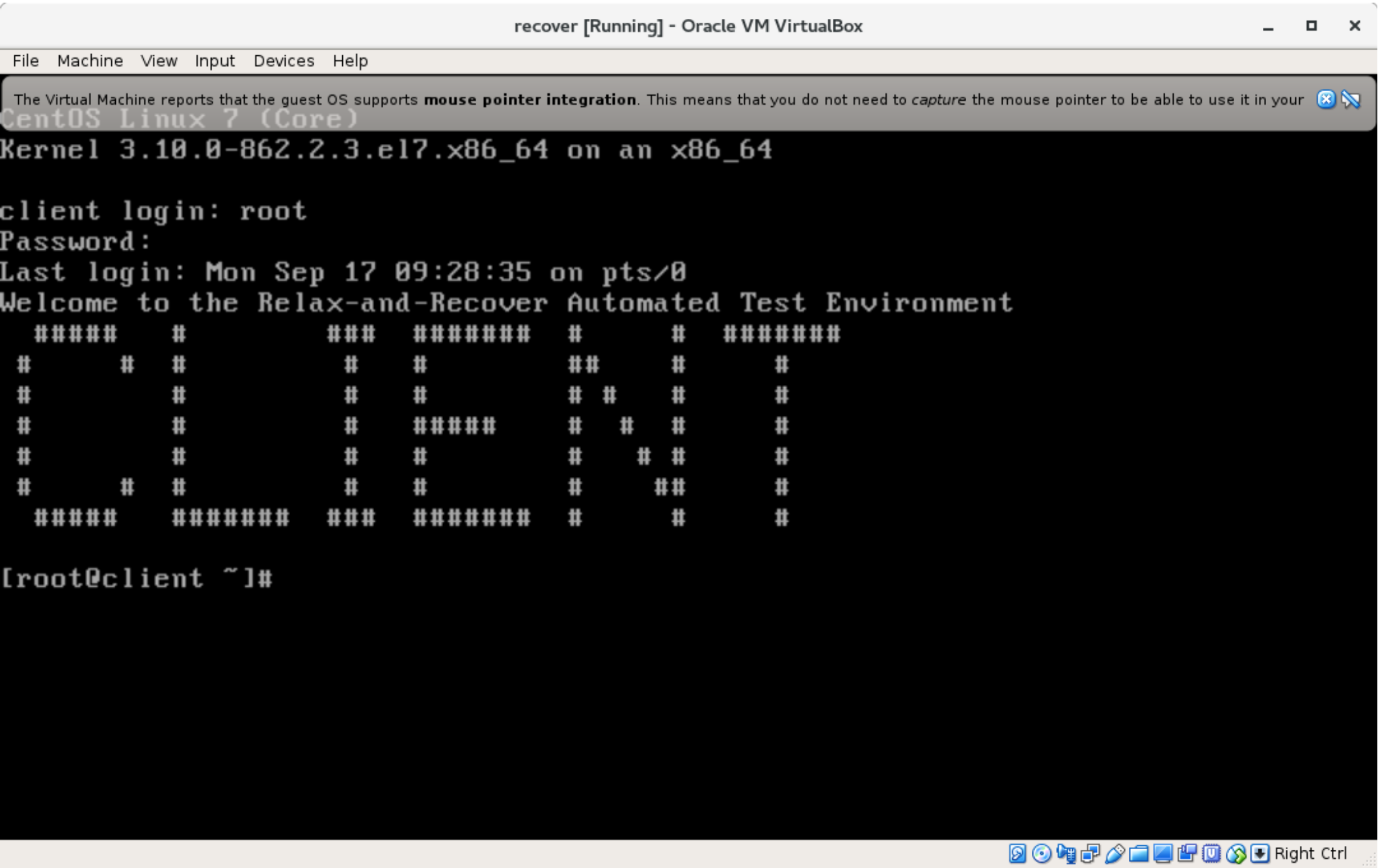

recover [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Relax-and-Recover v2.4
Recover client
Automatic Recover client
Other actions
Help for Relax-and-Recover
Boot First Local disk (hd0)
Boot Second Local disk (hd1)
Boot Next device
Hardware Detection Tool
ReBoot system
Power off system
Press [Tab] to edit, [F2] for help, [F1] for version info
Automatic boot in 5 seconds...
Rescue image kernel 3.10.0-862.2.3.el7.x86_64 Mon, 17 Sep 2018 09:29:38 +0200
BACKUP=BAREOS OUTPUT=ISO
Right Ctrl


recover [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
[ 0.237501] usbcore: registered new interface driver hub
[ 0.240418] usbcore: registered new device driver usb
[ 0.241148] PCI: Using ACPI for IRQ routing
[ 0.241788] NetLabel: Initializing
[ 0.242331] NetLabel: domain hash size = 128
[ 0.242836] NetLabel: protocols = UNLABELED CIPSOv4
[ 0.243120] NetLabel: unlabeled traffic allowed by default
[ 0.243656] Switched to clocksource refined-jiffies
[ 0.247567] pnp: PnP ACPI init
[ 0.248076] ACPI: bus type PNP registered
[ 0.249059] pnp: PnP ACPI: found 2 devices
[ 0.249578] ACPI: bus type PNP unregistered
[ 0.255960] Switched to clocksource acpi_pm
[ 0.256607] NET: Registered protocol family 2
[ 0.257134] TCP established hash table entries: 8192 (order: 4, 65536 bytes)
[ 0.257747] TCP bind hash table entries: 8192 (order: 5, 131072 bytes)
[ 0.258218] TCP: Hash tables configured (established 8192 bind 8192)
[ 0.258573] TCP: reno registered
[ 0.259023] UDP hash table entries: 512 (order: 2, 16384 bytes)
[ 0.259585] UDP-Lite hash table entries: 512 (order: 2, 16384 bytes)
[ 0.260852] NET: Registered protocol family 1
[ 0.261040] pci 0000:00:00.0: Limiting direct PCI/PCI transfers
[ 0.261638] pci 0000:00:01.0: Activating ISA DMA hang workarounds
[ 0.263031] Unpacking initramfs...
Right Ctrl


recover [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
The Virtual Machine reports that the guest OS supports mouse pointer integration. This means that you do not need to capture the mouse pointer to be able to use it in your
Start system layout restoration.
Creating partitions for disk /dev/sda (msdos)
Creating LVM PV /dev/sda3
Restoring LVM VG 'VolGroup00'
Sleeping 3 seconds to let udev or systemd-udevd create their devices...
Creating filesystem of type xfs with mount point / on /dev/mapper/VolGroup00-LogVol00.
Mounting filesystem /
Creating filesystem of type xfs with mount point /boot on /dev/sda2.
Mounting filesystem /boot
Creating swap on /dev/mapper/VolGroup00-LogVol01
Disk layout created.
waiting for job to start
waiting for job to finish
Restore job finished.
Please verify that the backup has been restored correctly to '/mnt/local'
in the provided shell. When finished, type exit in the shell to continue
recovery.
Bareos restore finished.
Recreating directories (with permissions) from /var/lib/rear/recovery/directories_permissions_owner_group
Running mkinitrd...
Right Ctrl


recover [Running] – Oracle VM VirtualBox
File Machine View Input Devices Help
iPXE (PCI C8:00.0) starting execution...ok
iPXE initialising devices...ok
iPXE 1.0.0+ -- Open Source Network Boot Firmware -- http://ipxe.org
Features: DNS TFTP HTTP PXE PXEXT Menu
net0: 08:00:27:40:34:54 using 82540em on PCI00:03.0 (open)
[Link:up, TX:0 TXE:0 RX:0 RXE:0]
DHCP (net0 08:00:27:40:34:54)....
Right Ctrl


recover [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
The Virtual Machine reports that the guest OS does not support mouse pointer integration in the current video mode. You need to capture the mouse (by clicking over the VM
CentOS Linux (3.10.0-862.2.3.el7.x86_64) 7 (Core)
Use the ↑ and ↓ keys to change the selection.
Press 'e' to edit the selected item, or 'c' for a command prompt.
The selected entry will be started automatically in 1s.
Right Ctrl


recover [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
The Virtual Machine reports that the guest OS supports mouse pointer integration. This means that you do not need to capture the mouse pointer to be able to use it in your
CentOS Linux 7 (Core)
Kernel 3.10.0-862.2.3.el7.x86_64 on an x86_64
client login: root
Password:
Last login: Mon Sep 17 09:28:35 on pts/0
Welcome to the Relax-and-Recover Automated Test Environment
[root@client ~]#
Right Ctrl


**You might consider to run, when the client VM was recovered, the following command:**

inspec exec ./inspec/compliance-checks -i ./insecure_keys/vagrant.private -t ssh://root@client | dos2unix -f | tee /export/rear-tests/logs/2018-09-17_09-28-19/inspec_results_client_after_recovery

Profile: InSpec Profile (compliance-checks)
Version: 0.1.0
Target:  ssh://root@client:22

✔ kernel.shmall: kernel.shmall check
  ✔ Kernel Parameter kernel.shmall value should eq 2097152
✔ kernel.shmmax: kernel.shmmax check
  ✔ Kernel Parameter kernel.shmmax value should eq 134217728
✔ fs.file-max: fs.file-max check
  ✔ Kernel Parameter fs.file-max value should eq 65536
✔
✔ filesystem-root: Verify / directory
  ✔ File / should be directory
✔ home-vagrant-exists: Verify /home/vagrant directory
  ✔ File /home/vagrant should be directory


× **iputils integrity: RPM integrity test on iputils package (1 failed)**
  ✔ System Package iputils should be installed
  × Command rpm -V iputils stdout should eq ""

  expected: ""
  got: ".....P  /usr/bin/ping\n.....P  /usr/sbin/arping\n.....P /usr/sbin/clockdiff\n"

  (compared using ==)

  Diff:
  @@ -1 +1,4 @@
  +.....P  /usr/bin/ping
  +.....P  /usr/sbin/arping
  +.....P  /usr/sbin/clockdiff

✔ root-account: The super user account
  ✔ User root should exist

Profile Summary: 9 successful controls, 1 control failure, 0 controls skipped
Test Summary: 23 successful, 1 failure, 0 skipped


# Contact

- Gratien D'haese
  e-mail: gratien.dhaese@it3.be
- web: http://www.it3.be
- Relax-and-Recover main project site:
  http://relax-and-recover.org/
- Relax-and-Recover Sources and Issues:
  https://github.com/rear/rear
- Commercial Support:
  http://www.it3.be/rear-support/