@schlomo @schlomo
@schlomoschapiro @schlomo@floss.social

TEKTIT CONSULTING

bareos

Relax and Recover: Automated Linux Recovery and Bare Metal Restore for Bareos

Schlomo Schapiro, Associate Partner / Principal Engineer, Tektit Consulting
27.03.2025, Bareos Expert Circle

Problem To Solve

Fully & End–2–End Automated

Disaster Recovery / Bare Metal Restore

Zero Knowledge, Hands Off,

Repeatable, Scaleable,

Universal, It–Just–Works,

Testable & Provable Quality

Ask Yourself: Mean Time to Restore Service

- After deploying a bad software update or configuration?
  → On 1000 servers?

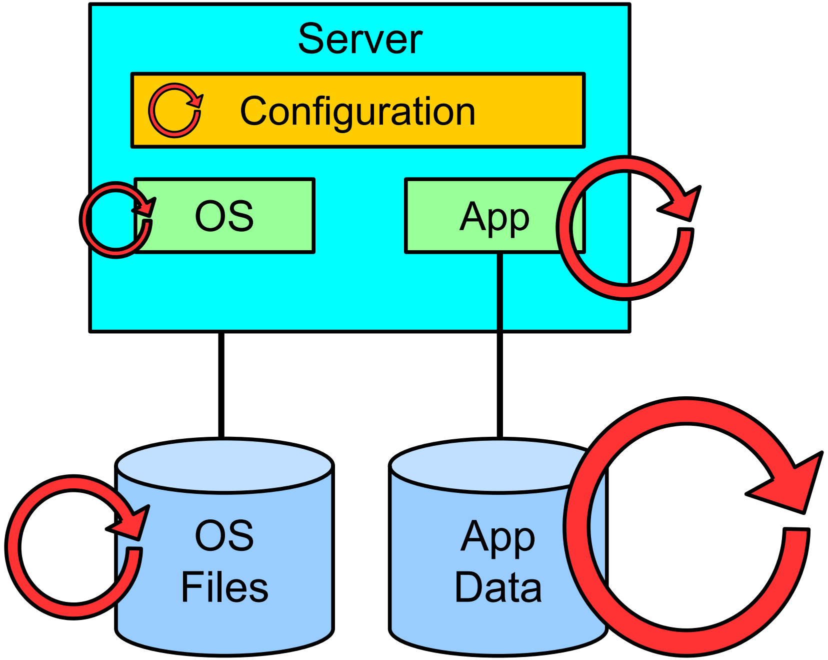
- After upgrading the Operating System to a faulty version?
  → On 50 servers? On 500 servers?

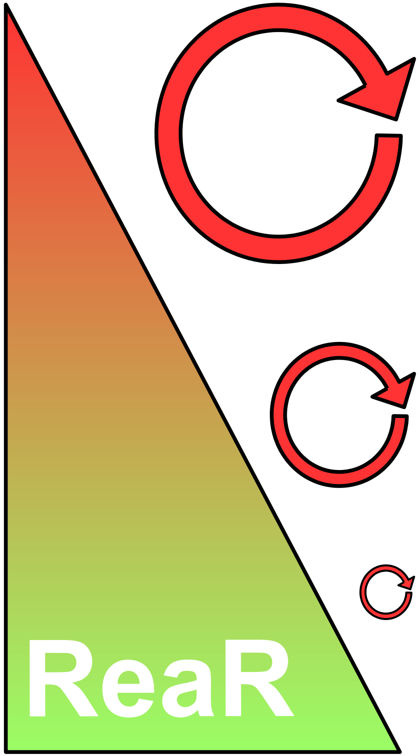
- After deleting the hard disk / SAN LUN of your main database?
  → After deleting 20 LUNs?

- After deleting the hard disk / SAN LUN of a hypervisor?
  → All the LUNs of a virtualization cluster?

- After flooding or burning the data center?
  → Getting a new data center?

Understanding Change Frequencies



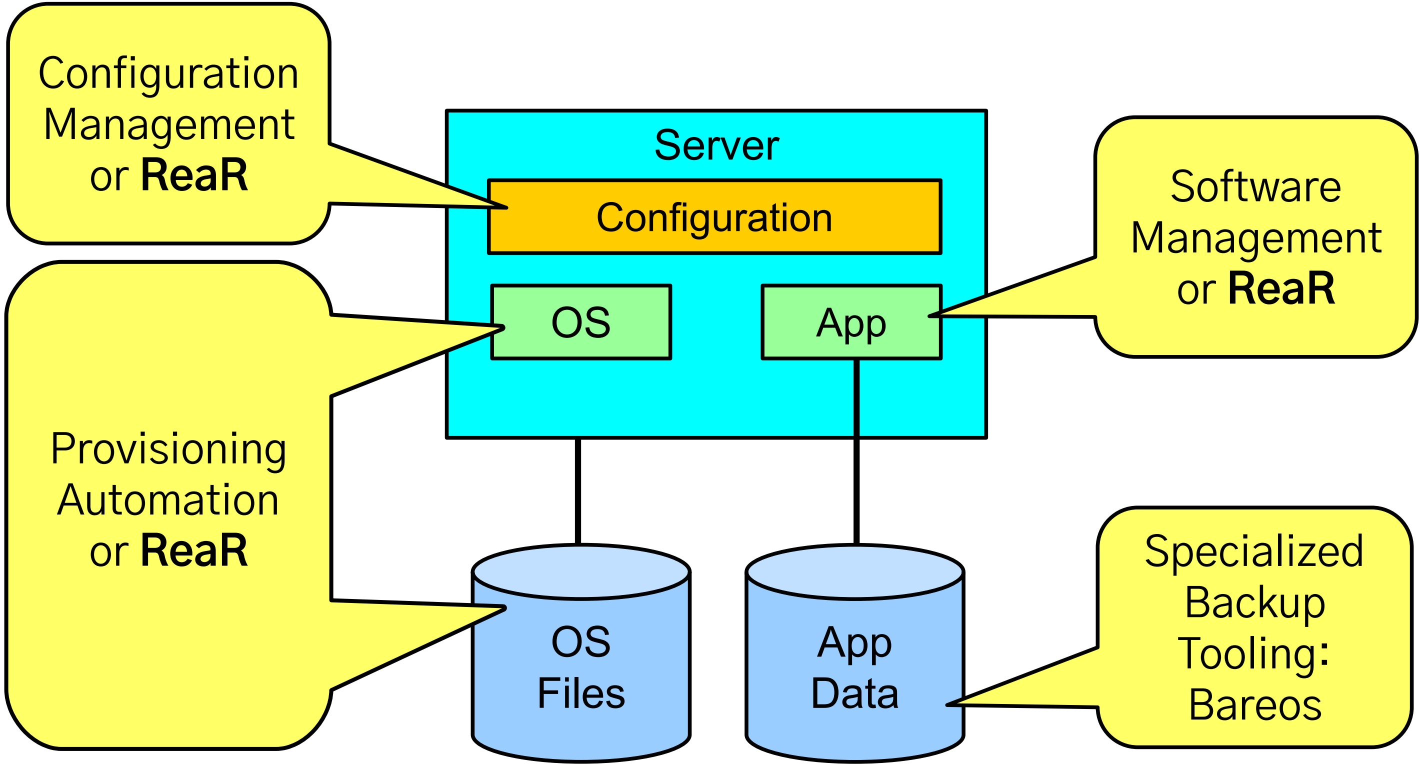


Constantly changing
Very high cost per
time lost

Regularly changing

Seldomly changing

Use the Best Tool for the Recovery / Restore Job


Configuration Management or **ReaR**
Server
Configuration
Software Management or **ReaR**
OS
App
Provisioning Automation or **ReaR**
OS Files
App Data
Specialized Backup Tooling: Bareos

Manual → Automated

Backup → Restore → Recovery

Relax and Recover

Why are Backups not Enough?

- “We have a backup of everything” – really?
- Backups of data are necessary – but it is only a starting point!
- Are not enough in case of losing the complete Operating System (OS)!
- Reinstalling the OS from scratch takes hours
- Restoring the backups takes more hours or days
- Fine–tuning of configurations takes days
- Even months later issues pop up!

→ Complete system backups provide much better protection!

Disaster Recovery Plan (DRP)

- DRP addresses need to recover from an emergency with minimum impact to the enterprise
- Protects enterprise from major services failure
- Minimizes risk to enterprise from delays in providing services
- Guarantees reliability of standby systems by testing and simulation
- Minimizes human decision-making required during disaster recovery

→ Automation creates a DRP that can scale out **and** up to cover everything!

Introducing Relax and Recover (ReaR)

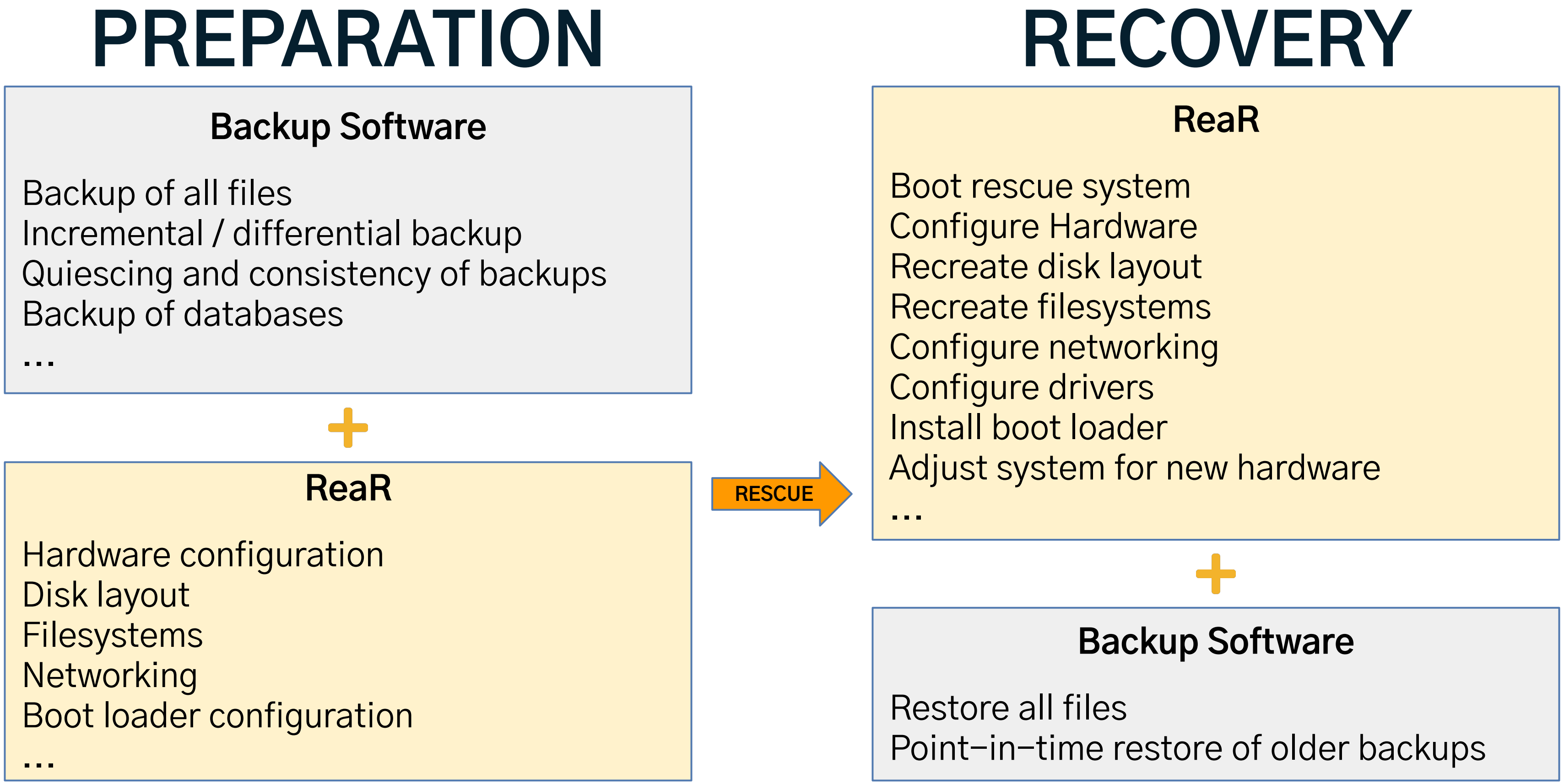
- 100% Open Source
- Integrates 17 backup solutions
- Goal:

  Fully automate everything related to Linux disaster recovery and bare metal restore

Basic ReaR Design Concepts

1. Every file on a Linux computer must be stored in a **Backup**
2. ReaR creates a bootable rescue media as **Output**
3. Rescue media boots on recovery computer
4. ReaR automates system recreation (disk layout, fileystems, networking …)
5. ReaR **facilitates** the **restore** of **all files from backup**
6. ReaR makes computer **bootable**
7. Reboot
8. Done ✅
9. 🎉 🥳 🍾 🎈 ❤️❤️❤️

ReaR specializes on Linux System Recovery



PREPARATION

Backup Software

Backup of all files
Incremental / differential backup
Quiescing and consistency of backups
Backup of databases
…

+

ReaR

Hardware configuration
Disk layout
Filesystems
Networking
Boot loader configuration
…

RESCUE

RECOVERY

ReaR

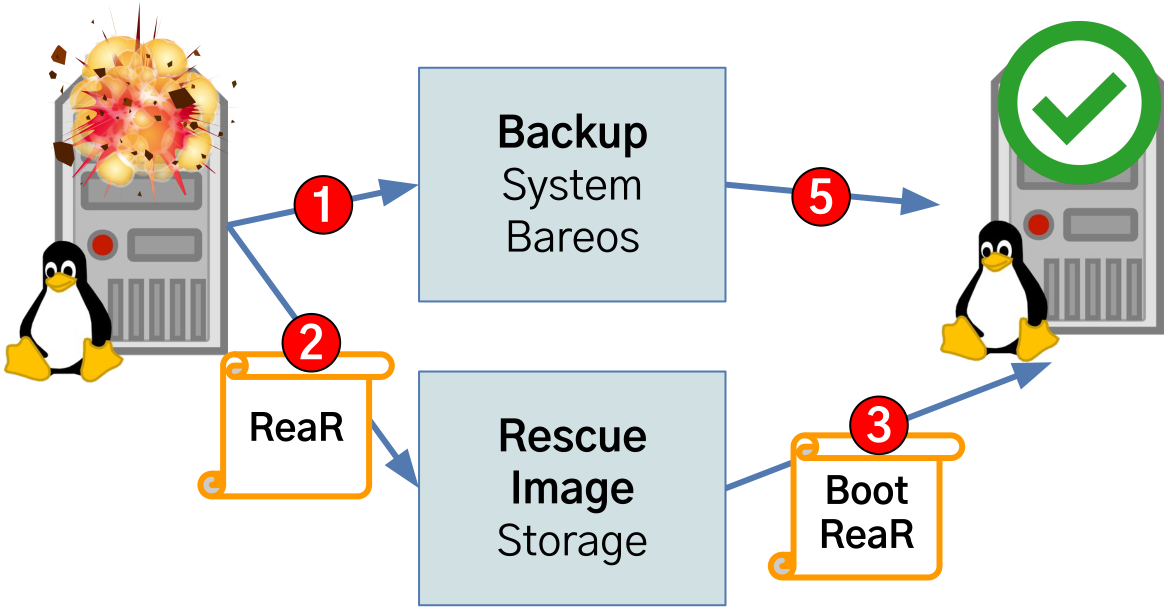
Boot rescue system
Configure Hardware
Recreate disk layout
Recreate filesystems
Configure networking
Configure drivers
Install boot loader
Adjust system for new hardware
…

+

Backup Software

Restore all files
Point-in-time restore of older backups

Disaster Recovery with ReaR



1

Backup System Bareos

2

ReaR

Rescue Image Storage

5

3

Boot ReaR

4

ReaR Rescue System:

Same name/IP/login as original system 😁

1. Configure disks
2. Format filesystems
3. Mount filesystems
4. Restore files 5
5. Install boot loader
6. Adjust system configuration
7. Reboot
8. Done

6

Disaster Recovery - Media

- Most important: External storage!
- Bootable media:
  - CD/DVD/ISO image
  - USB storage
  - PXE boot
- Separation between **boot media** and **backup data**
  - Boot the system from a (small) USB storage, (virtual) CD/DVD or LAN
  - Recover the system with backup software like Bareos ...

Usage of rear

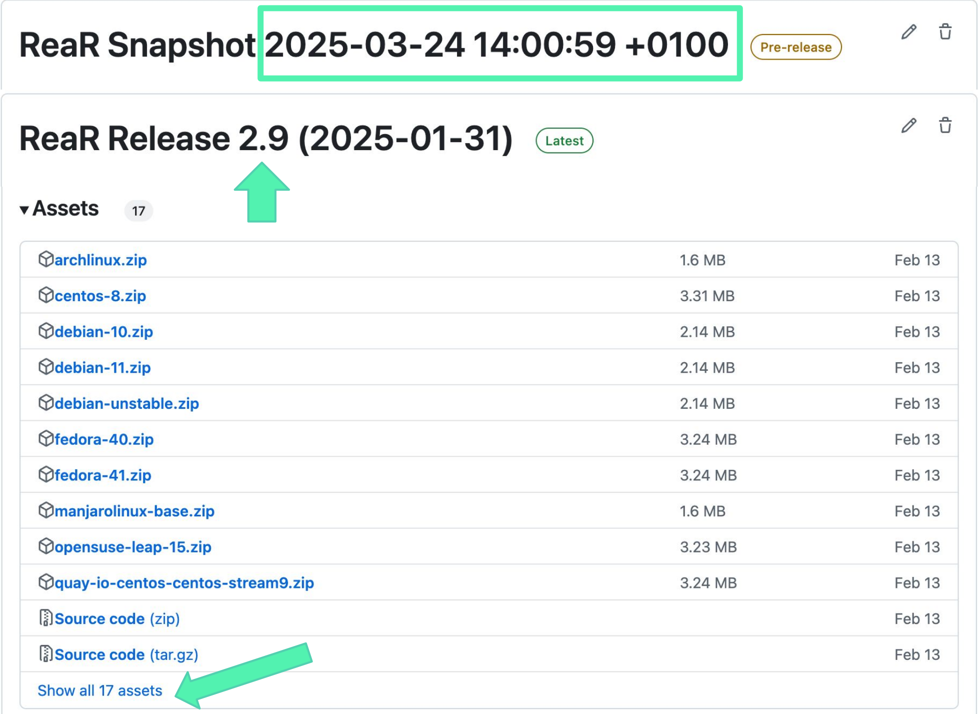
- Usage: `rear <Options> <Workflow>`
- Options:
  `-v, -d, -D --expose-secrets, --non-interactive ...`
- Workflows:
  - **`mkrescue`** (make rescue image)
  - **`mkbackup`** (create backup and rescue image)
  - **`recover`** (the actual recovery part, run from ReaR rescue system)
  - **`dump`** (show configuration)
  - more for special use cases:
    checklayout, udev, `format, mkopalpba, opaladmin, shell, validate`...

Installing ReaR

1. From your distro (old):
   a. `apt install rear`
   b. `dnf install rear`
   c. `zypper install rear`
   d. …

Installing ReaR

1. From your distro (old):
   a. `apt install rear`
   b. `dnf install rear`
   c. `zypper install rear`
   d. ...
2. From upstream (current development):
   a. github.com/rear/rear/releases/
   b. built automatically
   c. OS packages



ReaR Snapshot 2025-03-24 14:00:59 +0100 Pre-release

ReaR Release 2.9 (2025-01-31) Latest

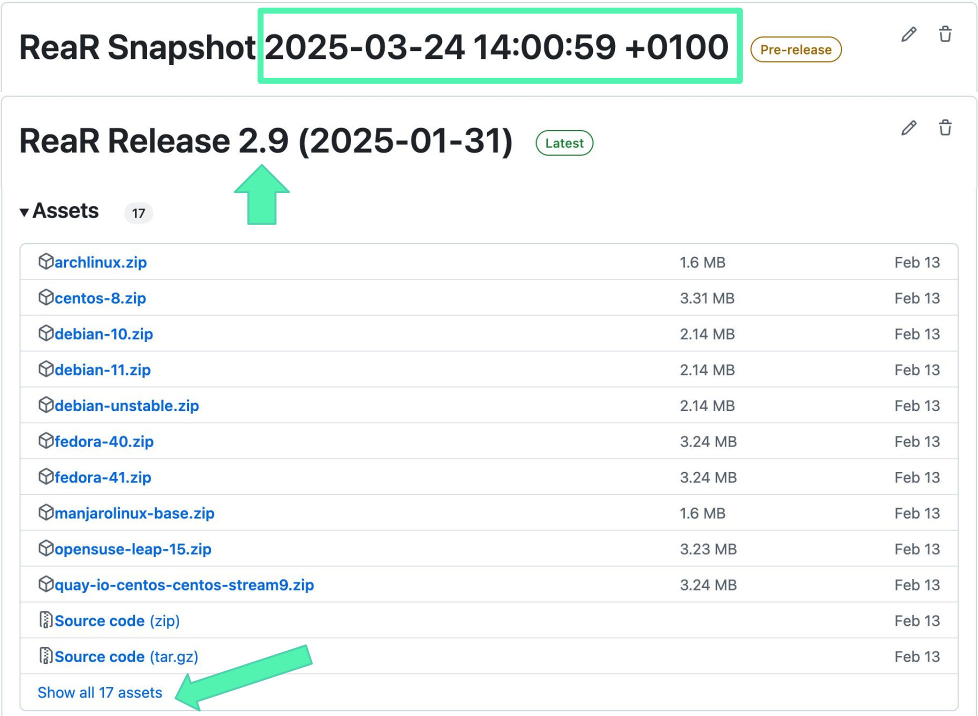
▾Assets 17

archlinux.zip	1.6 MB	Feb 13
centos-8.zip	3.31 MB	Feb 13
debian-10.zip	2.14 MB	Feb 13
debian-11.zip	2.14 MB	Feb 13
debian-unstable.zip	2.14 MB	Feb 13
fedora-40.zip	3.24 MB	Feb 13
fedora-41.zip	3.24 MB	Feb 13
manjarolinux-base.zip	1.6 MB	Feb 13
opensuse-leap-15.zip	3.23 MB	Feb 13
quay-io-centos-centos-stream9.zip	3.24 MB	Feb 13
Source code (zip)		Feb 13
Source code (tar.gz)		Feb 13

Show all 17 assets

Installing ReaR

1. From your distro (old):
   a. `apt install rear`
   b. `dnf install rear`
   c. `zypper install rear`
   d. ...
2. From upstream (current development):
   a. github.com/rear/rear/releases/
   b. built automatically
   c. OS packages
3. From source:
   a. github.com/rear/rear
   b. `make install`
4. *Use from source (portable):*
   `./usr/src/rear`



ReaR Snapshot 2025-03-24 14:00:59 +0100 Pre-release

ReaR Release 2.9 (2025-01-31) Latest

▾ Assets 17

archlinux.zip	1.6 MB	Feb 13
centos-8.zip	3.31 MB	Feb 13
debian-10.zip	2.14 MB	Feb 13
debian-11.zip	2.14 MB	Feb 13
debian-unstable.zip	2.14 MB	Feb 13
fedora-40.zip	3.24 MB	Feb 13
fedora-41.zip	3.24 MB	Feb 13
manjarolinux-base.zip	1.6 MB	Feb 13
opensuse-leap-15.zip	3.23 MB	Feb 13
quay-io-centos-centos-stream9.zip	3.24 MB	Feb 13
Source code (zip)		Feb 13
Source code (tar.gz)		Feb 13

Show all 17 assets

Configuring ReaR

- `/etc/rear/local.conf` → use for manual configuration
- `/etc/rear/site.conf` → use for configuration management
- `/usr/share/rear/conf/default.conf` → defaults & explanations
- We use Bash arrays, e.g.

  ```
  PROGS+=( mc )
  COPY_AS_IS+=( /{etc,usr/lib,usr/share}/mc )
  ```
- Configuration files are Bash scripts – use your fantasy
- Supply additional configuration files via `-C` option

Four Core Configuration Variables

1. BACKUP variable defines the “backup” method:
   BAREOS, NETFS, RSYNC …
2. BACKUP_URL variable defines the location where to store the backup archive (for internal backup methods only)
3. OUTPUT variable defines the “output” method:
   ISO, PXE, OBDR, USB, RAWDISK
4. OUTPUT_URL variable defines the location where to store the rescue image (ISO image, pxe configuration, extlinux configuration)

Check `default.conf` for details of all other configuration variables

ReaR Configuration Example - Bareos Production

```
BACKUP=BAREOS


# default approach is to store ISO within
# Bareos and pull ISO from backup when needed
# if rescue should be stored indepently
# OUTPUT_URL=ftp://server.domain/dr
# if Bareos auto-configuration fails
# BAREOS_CLIENT=client1-fd
# BAREOS_FILESET=LinuxAll
# BAREOS_RESTORE_JOB=RestoreFiles
```

Bareos Configuration

Bareos client configuration

- ReaR requires bconsole
- bconsole should be configured, that the system can only access its own data

Configuration on the Bareos Client:

```
/etc/bareos/bconsole.conf:

Director {
   Name = bareos-dir
   address = 192.168.8.123
}
Console {
   Name = "client1-console"
   Password = "tC=i@X})30X5"
}
```

Configuration on the Bareos Director:

```
Console {
    Name                   = "client1-console"
    Password               = "tC=i@X})30X5"
    Client ACL             = "client1-fd"
    Profile                = "rear-client-profile"
}
```

Live Demo

ReaR Configuration Example - ReaR Development

```
OUTPUT_URL=nfs://server.domain/srv/scratch
OUTPUT=PXE

BACKUP=NETFS
BACKUP_URL=$OUTPUT_URL

FIRMWARE_FILES=('no')
MODULES=('loaded_modules')
EXCLUDE_MD5SUM_VERIFICATION=all

AUTOSHRINK_DISK_SIZE_LIMIT_PERCENTAGE=80
```

Disaster Recovery for your Laptop (or Solo Server)

- Configuration:

```
BACKUP=NETFS
BACKUP_URL=usb:///dev/disk/by-label/REAR-000
USB_DEVICE_FILESYSTEM_LABEL=REAR-000
OUTPUT=USB
UDEV_WORKFLOW=mkbackup
```

- Create bootable recovery media on USB storage:

```
rear format /dev/sdb
```

- Automatically trigger backup via udev rule:

```
ACTION=="add", SUBSYSTEM=="block", ENV{ID_FS_LABEL}=="REAR-000", RUN+="/usr/sbin/rear udev"
```

- Insert USB storage and wait till system beeps 📣 to signal “done”

Advanced ReaR Configuration Options

- `COPY_AS_IS, PROGS`
  add custom files and programs
- `OUTPUT=PORTABLE`
  portable mode rescue image (~5 MB)
- `/etc/rear/mappings`
  pre-seed answers for changed hardware, e.g. network cards or host bus adapters, to facilitate an automated P2V or V2V mass migration
- `POST_RECOVERY_SCRIPT, PRE_RECOVERY_COMMANDS …`
  add your own hooks to run custom code before or after recovery or backup
- `USE_DHCLIENT=yes`
  force DHCP on rescue system instead of static IP from original system
- `TIMESYNC=`
  activate time sync in rescue system (useful for hardware)
- `SSH_ROOT_PASSWORD=, TTY_ROOT_PASSWORD`
  activate password login for SSH or for local login in rescue system

ReaR Internals

- **Bash** framework using Bash v4 features
- Main program: `/usr/sbin/rear`
- Configuration: `/usr/share/rear/conf/` and `/etc/rear/`
- Functions: `/usr/share/rear/lib/*-functions.sh`
- Workflows: `/usr/share/rear/lib/*-workflow.sh`
- Stages: `/usr/share/rear/*`
- Multi-dimensional script merging by `ARCH, OS, OS_VERSION, BACKUP, OUTPUT, BACKUP/OUTPUT` and more.
  Use `rear -s` (simulate) to see how it works.
- `tools/run-in-docker.sh` – run ReaR code in Docker on multiple distros

Relax, and Recover! – With Bareos!

No Backup? No Mercy!

🌐 relax-and-recover.org

Q&A — How may I help you?

tkt.dev/schlomo

We are not consultants. We are Partners, Coaches, Humans, Enablers, Catalysts, Sparring Partners, Experts … and sometimes a little annoying.

I focus on IT strategy, IT governance, technology and architecture management, security and compliance automation, related organisational changes, business continuity, open source and cloud technologies – and I'm available as a Principal Engineer or Technical Product Owner for short-term / interim support.

Examples:

- **Business-IT alignment & leveraging**, developing required skills and abilities for 21st century IT, leverage AI
- **SaaS compliance & governance**, data possession vs. ownership, IAM, integrations, backup & DR, shadow IT
- **Compliance Automation**, finding the "golden path" to a "golden state"
- **Secrets Management** for Datacenter, Cloud Infrastructure, IaaS/PaaS/SaaS
- **Open Source**, from usage to contribution, writing policies, using SBOM, establishing Open Source Stewardship
- **Good Engineering Practices**, GitOps, test driven development, good architecture decisions, known tech strategy
- **Business Continuity and Disaster Recovery** for office, Cloud infrastructure, data center & SaaS, with quality assurance, emergency communication & collaboration, hot & cold standby, no-restore solution, ransomware protection, Linux Disaster Recovery / Bare Metal Restore with "Relax and Recover (rear)" Open Source tooling

schlomo@tkt.dev

TEKTIT CONSULTING