CS615 - Aspects of System Administration

Backup, Monitoring

Department of Computer Science
Stevens Institute of Technology
Jan Schaumann
jschauma@stevens.edu
https://stevens.netmeister.org/615/

Backups vs. Restores

Start two instances:

- NetBSD (`ami-569ed93c`)
- OmniOS (`ami-0a01a5636f3c4f21c`)

Backups vs. Restores

Backups are boring.

Backups are tedious.

Nobody likes doing backups.

Backups vs. Restores

Backups are just a *means*
to accomplish a specific *goal*:

To have the ability to restore data.

Basic Terminology, Concepts, and Considerations

- ”full backup”
- ”incremental backup”
- ”synthetic backup” (Green Team link: `https://is.gd/0bKE1c`)
- file level vs. block level
- differential backup
- journalling vs. snapshots
- live data / open files, meta data (e.g., file- and filesystem), file data
- Recovery Point Objective (RPO)
- Recovery Time Objective (RTO)
- Business Continuity Plan (BCP)
- replaceable vs irreplaceable systems

Data Storage Media

What media can we back up to?

Data Storage Media

Media:

- magnetic tape
- traditional hard disk
- solid-state drive
- optical storage
- the cloud, why not

Data Storage Media

What factors do we have to consider when choosing a backup medium?

Data Storage Media

Factors:

- I/O performance (both read/write, sequential vs. random access, ...)
- reusability and degredation
- longevity
- mobility
- data integrity assurance (e.g., WORM - write once, read many)
- data compression, encryption
- deduplication
- availability

Long-term storage

- *full* set of level 0 backups
- separate set from regular backups
- usually stored off-site
- recovery / retrieval takes time
- limited granularity
- storage media considerations
- storage media transport considerations
- backup encryption and recovery key management

Backups and Restore Basics

When do we need backups?

- long-term storage / archival
- recover from data loss due to...

Backups and Restore Basics

When do we need backups?

- long-term storage / archival
- recover from data loss due to
  - equipment failure
  - user failure
  - natural disaster
  - security breach
  - software bugs

Backups and Restore Basics

When do we need backups?

- long-term storage / archival
- recover from data loss due to
  - equipment failure
  - user failure
  - natural disaster
  - security breach
  - software bugs

Think of your backups as *insurance*: you invest and pay for it, hoping you will never need it.

Reasons for Restore Requests

- file recovery
- system recovery (full or partial loss of e.g. a single system)
- disaster recovery

Disaster Recovery

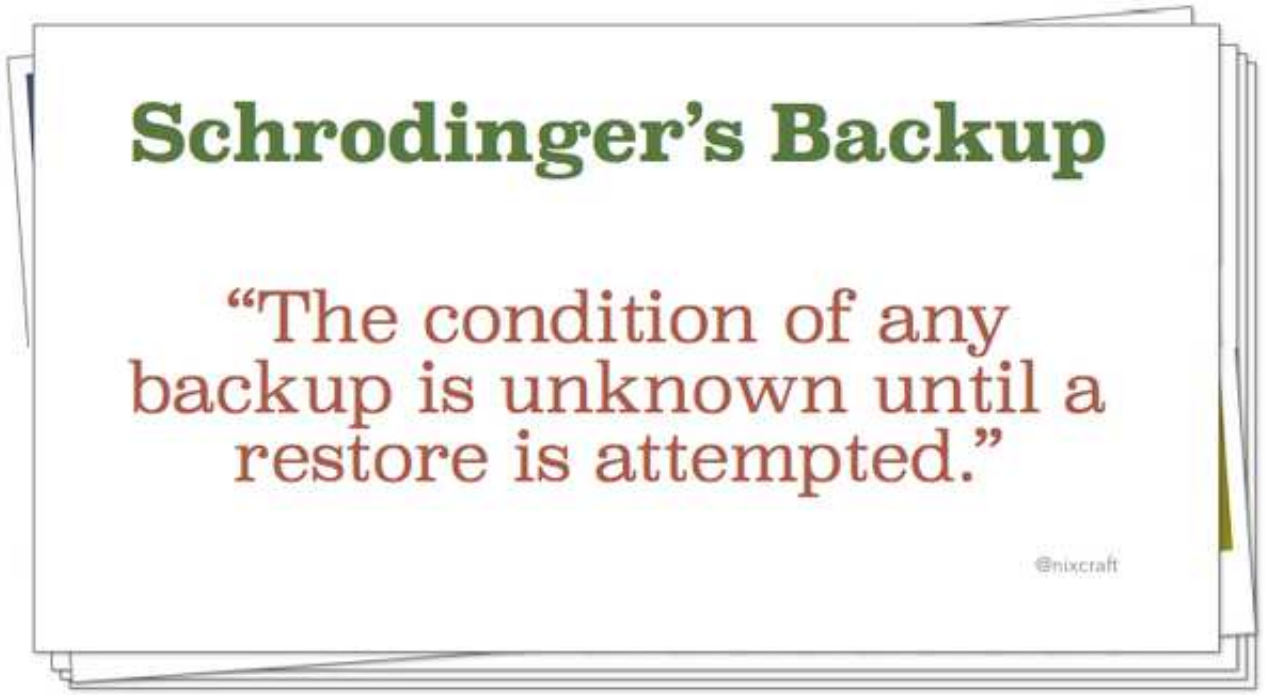
- loss of e.g. entire file system
- leads to downtime (of individual systems)
- RAID may help
- takes long time to restore
- may require retrieval of archival backups from long-term storage
- often involves *some* data loss
- 3-2-1 Rule:
  - keep at least 3 copies of your data
  - keep at least 2 copies on different storage media
  - keep at least 1 copy offsite

Disaster Recovery

- loss of e.g. entire file system
- leads to downtime (of individual systems)
- RAID may help
- takes long time to restore
- may require retrieval of archival backups from long-term storage
- often involves *some* data loss
- 3-2-1 Rule:
  - keep at least 3 copies of your data
  - keep at least 2 copies on different storage media
  - keep at least 1 copy offsite

Beware: disasters scale up much faster than your backup strategy!

To the backups!



Black Team link:`https://www.guru99.com/recovery-testing.html`

Trusting your backups

Backing up data requires superuser privileges!
Red Team link: CVE-2019-16155:
`https://danishcyberdefence.dk/blog/forticlient_linux`

A backup is a *copy* of the data. If the data is corrupt, your backup may become corrupt.

To restore data from a trusted backup, you can only use trusted tools.

Verify the authenticity of your backups!

Blue Team link: `https://is.gd/lG6ZQM`

File deletion recovery

Accidentally deleted files ought to be recoverable for a certain amount of time:

- ”Undo”
- time window and granularity requirements (Recovery Point Objective)
- restore time (Recovery Time Objective), including
  - actual time spent restoring
  - waiting until resources permit the restore
  - staff availability
- self-service restore

But note: sometimes people *do* want to delete data and it be gone!

Filesystem backup

```
ssh netbsd-instance "dump -u -0 -f - /" | bzip2 -c -9 >tmp/ec2.0.bz2
  DUMP: Found /dev/rxbd1a on / in /etc/fstab
  DUMP: Date of this level 0 dump: Mon Apr  2 19:34:30 2018
  DUMP: Date of last level 0 dump: the epoch
  DUMP: Dumping /dev/rxbd1a (/) to standard output
  DUMP: Label: none
  DUMP: mapping (Pass I) [regular files]
  DUMP: mapping (Pass II) [directories]
  DUMP: estimated 962609 tape blocks.
  DUMP: Volume 1 started at: Mon Apr  2 19:34:34 2018
  DUMP: dumping (Pass III) [directories]
  DUMP: dumping (Pass IV) [regular files]
  DUMP: 42.40% done, finished in 0:06
  DUMP: 83.38% done, finished in 0:01
  DUMP: 963445 tape blocks
  DUMP: Volume 1 completed at: Mon Apr  2 19:46:38 2018
  DUMP: Volume 1 took 0:12:04
  DUMP: Volume 1 transfer rate: 1330 KB/s
  DUMP: Date of this level 0 dump: Mon Apr  2 19:34:30 2018
  DUMP: Date this dump completed:  Mon Apr  2 19:46:38 2018
  DUMP: Average transfer rate: 1330 KB/s
  DUMP: level 0 dump on Mon Apr  2 19:34:30 2018
  DUMP: DUMP IS DONE
```

Filesystem backup

```
$ ssh netbsd-instance
netbsd$ cat /etc/dumpdates
/dev/rxbd1a      0 Mon Apr  2 19:34:30 2018
netbsd# mkdir -p /usr/local/data

$ scp -r some-data netbsd-instance:/usr/local/data/
$ ssh netbsd-instance "dump -u -i -f - /" | bzip2 -c -9 >tmp/ec2.1.bz2
  DUMP: Found /dev/rxbd1a on / in /etc/fstab
  DUMP: Date of this level i dump: Mon Apr  2 20:09:24 2018
  DUMP: Date of last level 0 dump: Mon Apr  2 19:34:30 2018
  DUMP: Dumping /dev/rxbd1a (/) to standard output
  DUMP: Label: none
  DUMP: mapping (Pass I) [regular files]
  DUMP: mapping (Pass II) [directories]
  DUMP: estimated 25307 tape blocks.
  DUMP: Volume 1 started at: Mon Apr  2 20:09:33 2018
  DUMP: dumping (Pass III) [directories]
  DUMP: dumping (Pass IV) [regular files]
  DUMP: 25244 tape blocks
  DUMP: Volume 1 completed at: Mon Apr  2 20:09:50 2018
  DUMP: Volume 1 took 0:00:17
  DUMP: Volume 1 transfer rate: 1484 KB/s
  DUMP: Date of this level i dump: Mon Apr  2 20:09:24 2018
  DUMP: Date this dump completed:  Mon Apr  2 20:09:50 2018
```

```
DUMP: Average transfer rate: 1484 KB/s
DUMP: level i dump on Mon Apr  2 20:09:24 2018
DUMP: DUMP IS DONE
```

Filesystem backup

```
netbsd# rm -fr /usr/local/data /etc/resolv.conf

$ bzip2 -d -c ec2.1.bz2 | ssh ec2-instance "cd /; /sbin/restore xf -"
$ bzip2 -d -c ec2.0.bz2 | ssh ec2-instance "cd /; /sbin/restore xf - etc/resolv.conf"
```

Poor Man's Cloud Backup via `tar(1)`

Copying to a file system:

```
$ tar cf - data/ | ssh ec2-instance "tar -xf - -C /var/backups/$(date)"
```

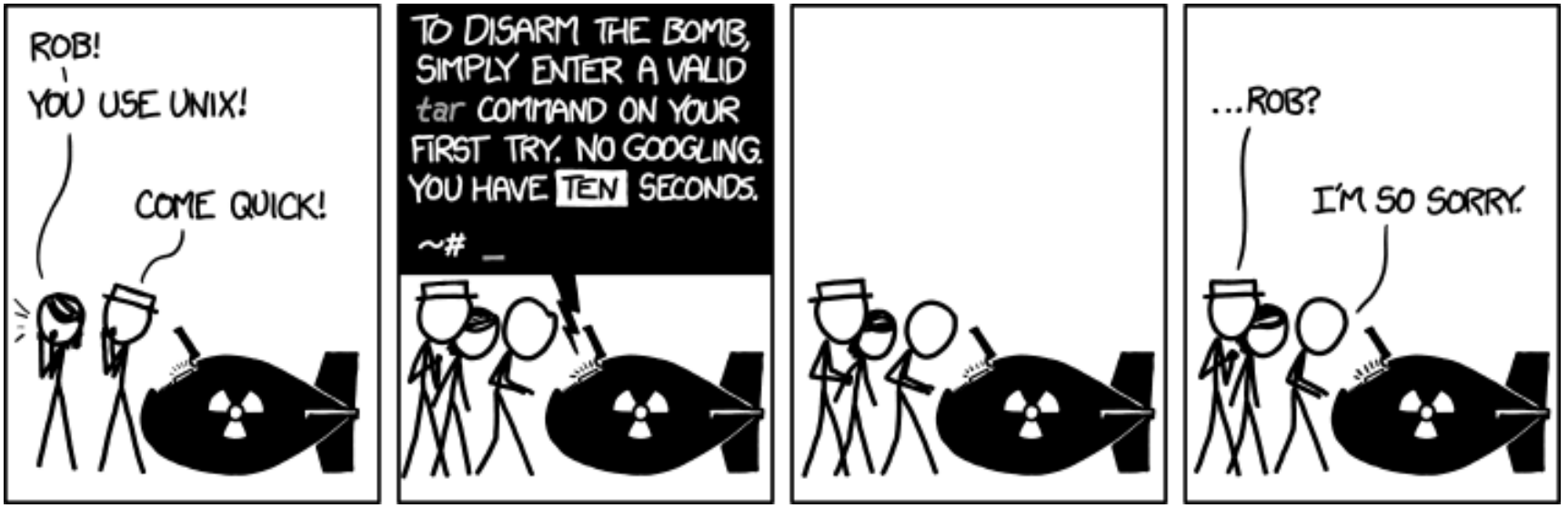
Writing to a block device, no filesystem necessary:

```
$ tar cf - data/ | ssh ec2-instance "dd of=/dev/rxb2a"
$ ssh ec2-instance "dd if=/dev/rxb2a" | tar tvf -
```

Encrypting along the way:

```
$ tar cf - data/ | gpg --encrypt -r recipient | ssh ec2-instance "dd of=/dev/rxb2a"
```

Know a Unix Command



https://www.xkcd.com/1168/
https://stevens.netmeister.org/615/tar.html

Filesystem considerations

Recall from Lecture 03 that our data can often be classified like so:

	shareable content	unshareable content
static data	`/usr` `/opt`	`/boot` `/etc`
variable data	`/var/mail` `/var/spool/news`	`/var/run` `/var/lock`

See also: `fstab(5)`

`https://stevens.netmeister.org/615/backup-exercise.html`

Backups vs. Snapshots

`dump(8)` preserves files (and file attribute), so that deletion of a file can be undone.

But what about intended file deletions in incremental backups?

Backups vs. Snapshots

`dump(8)` preserves files (and file attribute), so that deletion of a file can be undone.

But what about intended file deletions in incremental backups?

Enter `rsync(1)`:

```
$ rsync -e ssh -az remote:/. backup/.
$ ssh remote
remote# pkg_add whatever
remote# rm some files
remote# exit
$ rsync -e ssh -az --delete remote:/. backup/.
```

Backups vs. Snapshots

`dump(8)` preserves files (and file attribute), so that deletion of a file can be undone.

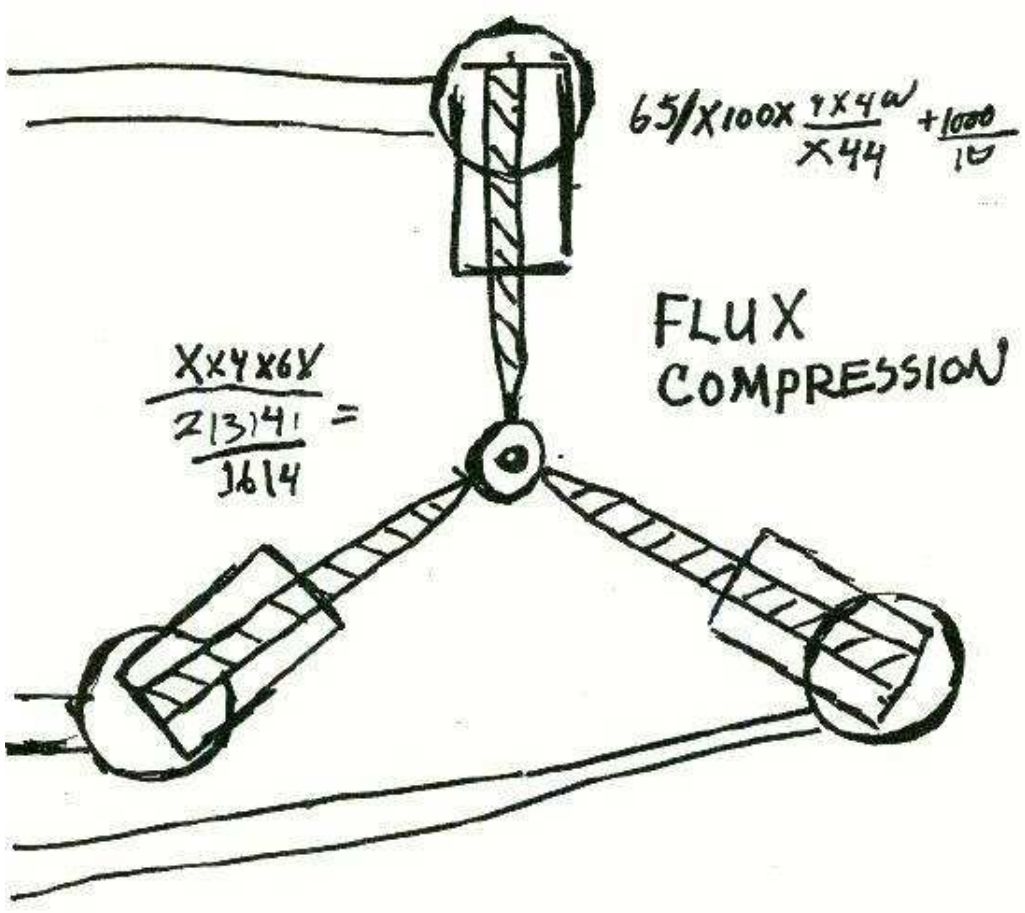
But what about intended file deletions in incremental backups?

Enter `rsync(1)`:

```
$ rsync -e ssh -az remote:/. backup/.
$ ssh remote
remote# pkg_add whatever
remote# rm some files
remote# exit
$ rsync -e ssh -az --delete remote:/. backup/.
```

But now we've lost the ability to restore a file we once intended to remove but then (much later) changed our mind about. Grrrr. I wish there was a way to simply go back in time to when the file still existed...

Filesystem backup



Filesystem backup

Filesystem backup

Filesystem backup

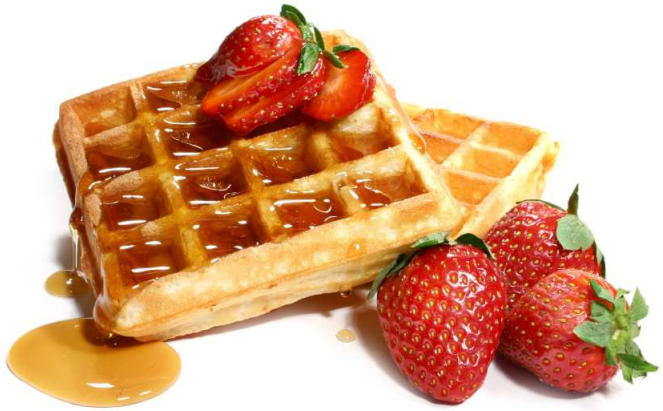
Example: Mac OS X “Time Machine”:

- automatically creates a full backup (equivalent of a ”level 0 dump”) to separate device or NAS, recording (specifically) last-modified date of all directories
- every hour, creates a full copy via *hardlinks* (hence no additional disk space consumed) for files that have not changed, new copy of files that have changed
- changed files are determined by inspecting last-modified date of directories (cheaper than doing comparison of all files’ last-modified date or data)
- saves hourly backups for 24 hours, daily backups for the past month, and weekly backups for everything older than a month.

Filesystem backup

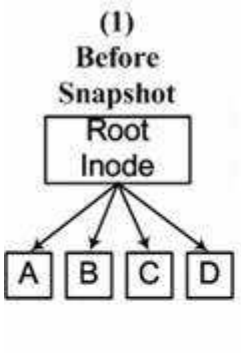
Example: WAFL (Write Anywhere File Layout)

- used by NetApp's "Data ONTAP" OS
- a snapshot is a read-only copy of a file system (cheap and near instantaneous, due to CoW)
- uses regular snapshots ("consistency points", every 10 seconds) to allow for speedy recovery from crashes



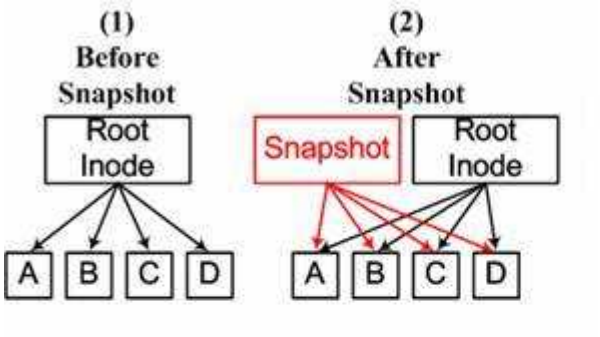
Filesystem backup

Example: WAFL (Write Anywhere File Layout)



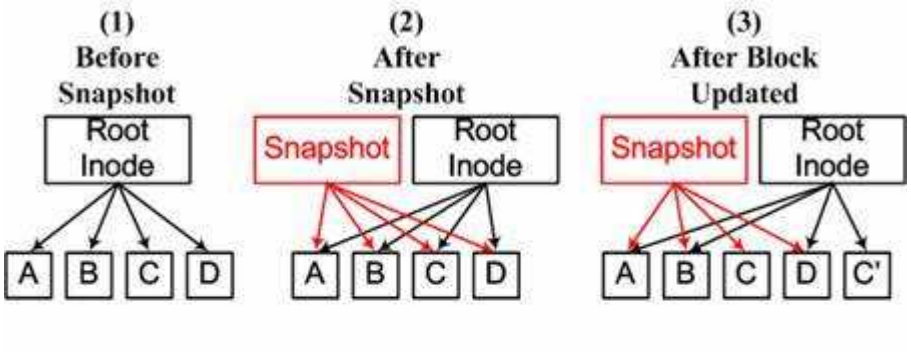
Filesystem backup

Example: WAFL (Write Anywhere File Layout)



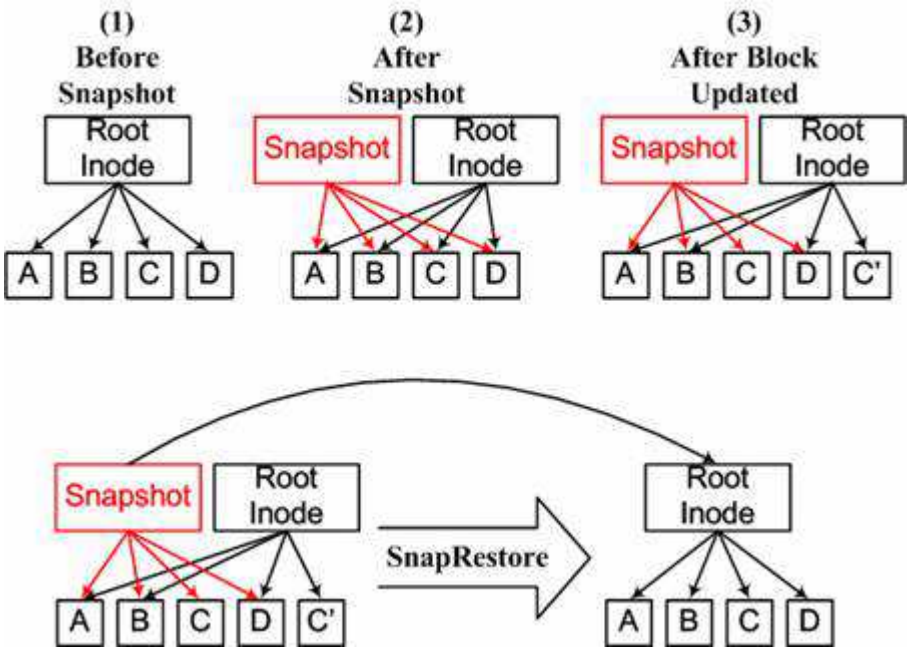
Filesystem backup

Example: WAFL (Write Anywhere File Layout)



Filesystem backup

Example: WAFL (Write Anywhere File Layout)



Filesystem backup

Example: ZFS snapshots

- ZFS uses a copy-on-write transactional object model (new data does not overwrite existing data, instead modifications are written to a new location with existing data being referenced), similar to WAFL
- a snapshot is a read-only copy of a file system (cheap and near instantaneous, due to CoW)
- initially consumes no additional disk space; the writable filesystem is made available as a “clone”
- conceptually provides a branched view of the filesystem; normally only the “active” filesystem is writable

ZFS Snapshots

```
$ start-omnios
$ ssh e2-instance
# zfs snapshot rpool/ROOT/omnios-r151030c@202004051830
# ls -la / | grep zfs
# ls -la /.zfs
total 6
dr-xr-xr-x   4 root     root           4 May  3  2018 .
drwxr-xr-x  23 root     root          24 Apr  8 14:33 ..
dr-xr-xr-x   2 root     root           2 May  3  2018 shares
dr-xr-xr-x   4 root     root           4 Apr  8 14:35 snapshot
# rm /root/.ssh/authorized_keys
# echo oh no > /root/file
```

ZFS Snapshots

Restoring individual files

```
# diff -bur /root/. /.zfs/snapshot/202004051830/root/.
Only in /root/.: .bash_history
Common subdirectories: /root/./.ssh and /.zfs/snapshot/201904081035/root/./.ssh
Only in /root/.: file
Only in /.zfs/snapshot/201904081035/root/./.ssh: authorized_keys
# cp /.zfs/snapshot/201904081035/root/./.ssh/authorized_keys /root/.ssh/authorized_ke
```

Rolling back:

```
# rm /root/.ssh/authorized_keys
# zfs rollback rpool/ROOT/omnios-r151030c@202004051830
# ls -l /root/.ssh/authorized_keys
-rw-------   1 root     root         389 Apr  8 15:19 /root/.ssh/authorized_keys
# ls /root/file
/root/file: No such file or directory
```

Summary

- backups are most commonly done as incrementals of a filesystem, mountpoint, or directory hierarchy
- consider (long-term) storage:
  - media and location
  - increased storage requirements
  - privacy and safety of the data
- self-service restores and filesystem snapshots
- backups need to be:
  - regular, frequent, automated
  - invisible
  - verifiable
  - regularly tested

Hooray!

5 minute break

Reading

Hurricane Sandy

- http://is.gd/aaxzvI
- http://is.gd/Y75pEA
- http://is.gd/32Az7y
- http://is.gd/FhAuFZ

Reading

Backups with `dump(8)` and `restore(8)`:

- `dump(8)` and `restore(8)`
- `https://is.gd/bXG9of`

Filesystem snapshots:

- `https://en.wikipedia.org/wiki/Snapshot_(computer_storage)`
- `https://en.wikipedia.org/wiki/Time_Machine_(Apple_software)`
- `http://comet.lehman.cuny.edu/jung/cmp426697/WAFL.pdf`

Book: `http://www.oreilly.com/catalog/unixbr/`

Reading