storage.franken.de

Datensicherung im Netz

Übersicht

- Hardware und Software
  - ZFS und Container
  - Firewall und Sicherheit
- Wie stellt sich das für den Nutzer dar
  - Offene Containerumgebung, eigenes backup-system
  - Beispiele für die Nutzung
  - Sonstige Spielregeln

Motivation

Warum storage.franken.de

- Eine Platform um offen mit “Cloud Backup” zu experimentieren
- Experimentieren im Sinne von
  - Verschiedene Protokolle und Mechanismen
  - Eigene Kontrolle der Nutzer darüber

Nicht “experimentell” ist der Umgang mit Daten selbst:

- Zuverlässigkeit und Redundanz für ernste Nutzung

Der Server

Dedicated Server mit

- 4x 10 TB HDD (7200rpm)
- 32GB RAM ECC
- CPU E3-1270 v3 @ 3.50GHz (4 cores)

Bei Hetzner in Falkenstein

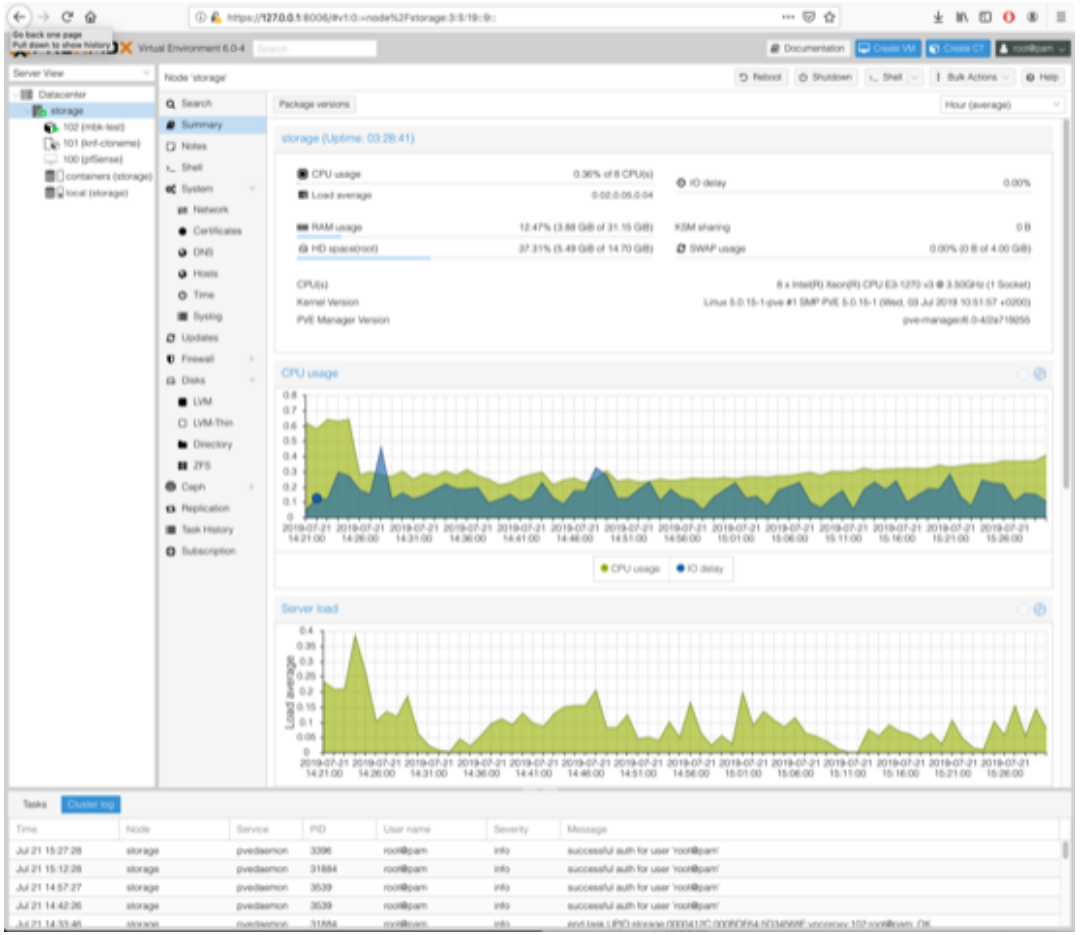
Warum keine eigene Hardware ?

- Wir wissen noch nicht wohin es mit dieser Platform geht
- Als Gäste/Mitnutzer bei DFN und den Unis wollen wir keinen Stress mit “TBs” verursachen

Proxmox und ZFS

Proxmox als Platform: Container und ZFS mit einfacher Oberfläche

- Pro Nutzer ein LXC Container mit Debian 10
- Der Host selbst wird nicht für andere Zwecke genutzt
- ZFS Volumes für LXC root und /data



Ein Wort zum Platz

ZFS mit RAIDZ1 (eine disk redundanz)

25 TiB freier Platz

Wir werden bei ca. 20 TiB Nutzung erweitern

Wenn ich Terabyte sage meine ich Tebibyte

```
ata1.00: ATA-10: TOSHIBA MG06ACA10TEY, 0103, max UDMA/133
ata1.00: 19532873728 sectors, multi 16: LBA48 NCQ (depth 32), AA
```

Dezimalpräfixe			Unterschied gerundet	Binärpräfixe gemäß IEC		
Name	Symbol	Anzahl Bytes[G 1]		Name	Symbol	Anzahl Bytes
Kilobyte	kB[G 2]	1 000 = 10^3	2,4 %	Kibibyte	KiB[G 3]	1 024 = 2^{10}
Megabyte	MB	1 000 000 = 10^6	4,9 %	Mebibyte	MiB	1 048 576 = 2^{20}
Gigabyte	GB	1 000 000 000 = 10^9	7,4 %	Gibibyte	GiB	1 073 741 824 = 2^{30}
Terabyte	TB	1 000 000 000 000 = 10^{12}	10,0 %	Tebibyte	TiB	1 099 511 627 776 = 2^{40}
Petabyte	PB	1 000 000 000 000 000 = 10^{15}	12,6 %	Pebibyte	PiB	1 125 899 906 842 624 = 2^{50}

Was bietet ZFS

Regelmässige ‘scrubs’ erkennen Fehler früher

```
root@storage ~ # zpool status
  pool: zpool
 state: ONLINE
  scan: scrub repaired 0B in 0 days 00:06:30 with 0 errors on Sun Nov 10 00:30:31 2019
config:

        NAME                                STATE     READ WRITE CKSUM
        zpool                               ONLINE       0     0     0
          raidz1-0                          ONLINE       0     0
            wwn-0x5000039938ca3dec-part4    ONLINE       0     0
            wwn-0x5000039938ca3dfe-part4    ONLINE       0     0
            wwn-0x5000039938ca3e09-part4    ONLINE       0     0
            wwn-0x5000039938ca3de5-part4    ONLINE       0     0

errors: No known data errors
```

root <root@qnas.axiros.com>

ZFS io error for qarchive on qnas

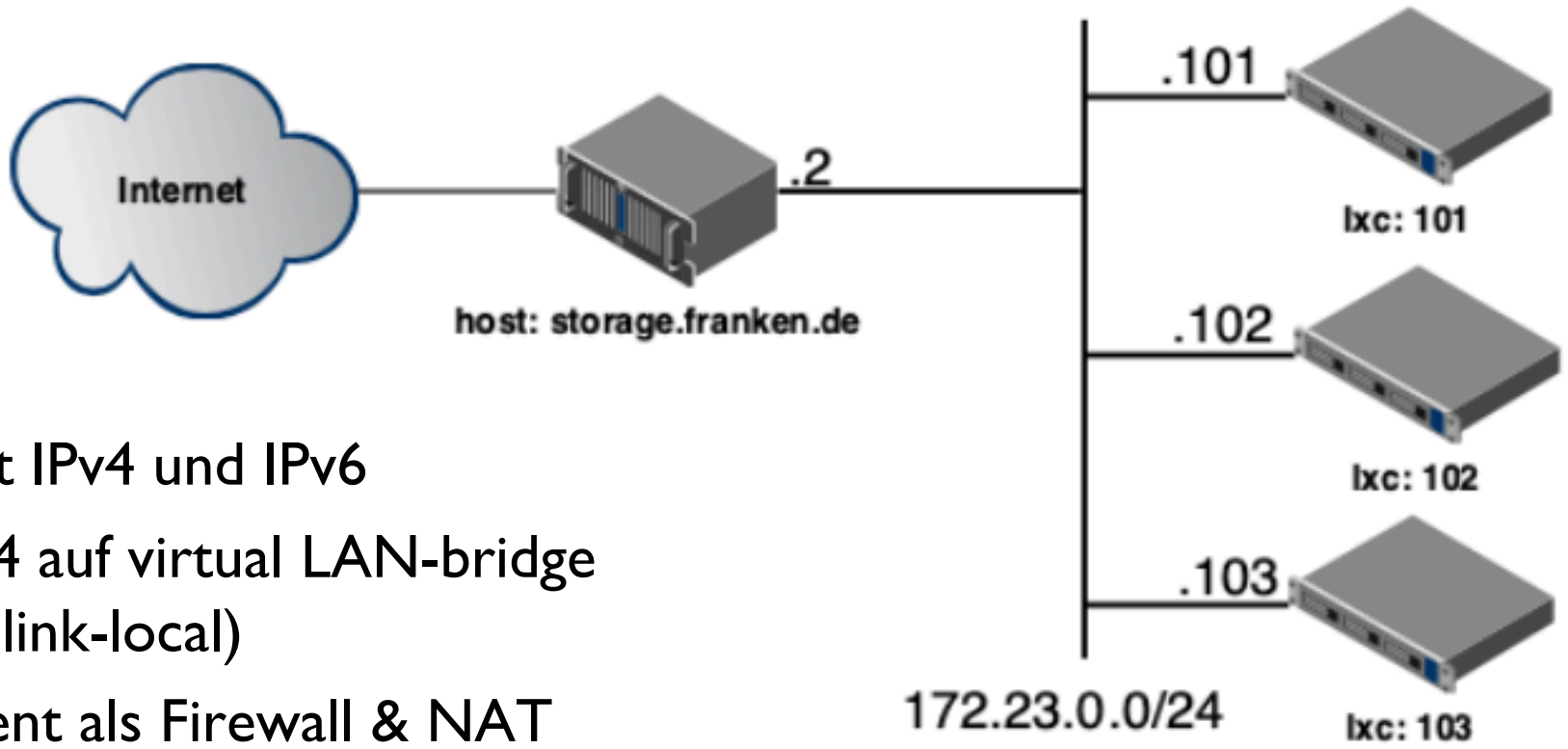
To: root <root@qnas.axiros.com>

```
ZFS has detected an io error:

  eid: 136
class: io
 host: qnas
 time: 2019-11-10 15:41:35+0100
vtype: disk
vpath: /dev/sda1
vguid: 0xEF14E4492A8D00B2
cksum: 0
 read: 0
write: 0
 pool: qarchive
```

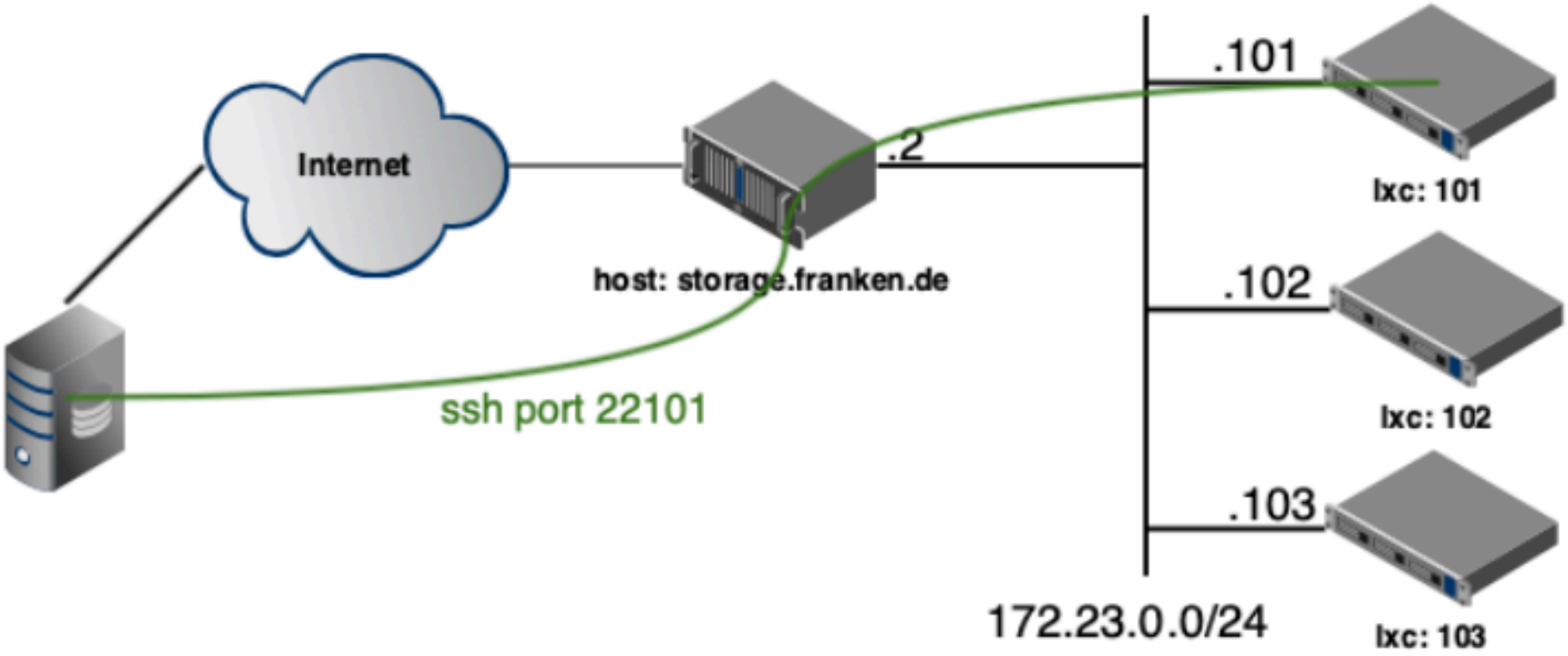
Noch offen: Bieten wir den Nutzern
automatische Snapshots der Daten ?

Das Netzwerk



- Host hat IPv4 und IPv6
- Nur IPv4 auf virtual LAN-bridge
  (IPv6 link-local)
- Host dient als Firewall & NAT

Das Netzwerk



Nur ssh und nur mit key authentication

Der Container

Pro Container:

- 8 GB root, 2 TB /data
- 4 GB RAM
- 2 vCPU

```
root@mbk-test:~# df -h
Filesystem                Size  Used Avail Use% Mounted on
zpool/subvol-102-disk-0   8.0G  2.0G  6.1G  25% /
zpool/subvol-102-disk-1   2.0T  108G  1.9T   6% /data
```

Container Nutzung I

Der Container fühlt sich fast wie eine VM in einem lokalen LAN an

- Netzwerkzugang von innen nach aussen ist offen via NAT
- Installation beliebiger Pakete
  - Bequem von Debian (`apt-get install borg-backup`)
  - Oder selbst gebaut (`git clone …; make; make install`)
- Installation neuer Dienste
  - Diese sind allerdings nur über ssh port-forward zu erreichen
- Im Container bist Du `root` und `reboot` tut was man erwartet

Wir erwarten dass ein “Datensicherungsdienst” läuft – welcher und wie genau bestimmt der Nutzer.

Container Nutzung II

Was geht NICHT:

- Direkte ZFS Interaktionen (zfs send/receive, snapshot etc).
- Andere Betriebsysteme als Linux
  - Heute auch nur Debian 10 (Buster, das aktuelle)
- Anderer Zugriff als ssh mit public key
  - Keine Passwörter um die Rate-Roboter fernzuhalten
  - Keine anderen Ports **von aussen** erreichbar
- Keine echten root-rechte auf dem Host
  - Der Container ist nicht "privileged", kernel-module manipulieren oder direkter device Zugang sind nicht möglich
- *"Container im Container" – insbesondere docker - lassen wir das zu ?*

Cloud Backup

"Datensicherungsdienst im Netz" … Cloud Backup

Kontext: Ein grosses (initiales) backup kann Wochen dauern

Anforderungen:

- Verschlüsselung auf Nutzerseite
- Kompression und Deduplizierung
- Robust gegen Netzunterbrechungen, unterbrochene backups fortsetzen
- Bandbreitenlimits, insbesondere beim upload
- Historie und Lebenszyklus
- Flexible Zeitpläne

Beispiele:

ssh-basiert

- borg-backup
- sftp + <?>
- Rsync + <?>

Cloud / Web

- S3 + <?>
- Seafile

Minio I

Minio (https://min.io/) ist ein Amazon S3 kompatibler Speicherdienst

Was ist S3 ?

- S3 steht für “Simple Storage Service“
- Ein “object store” Dienst den Amazon in seiner Cloud anbietet
- Diese API hat sich zum de-fakto Standard für Speicherdienste entwickelt. Es gibt dutzende Anbieter für Dienste und Clients.

Und hier wird es für uns interessant:

- Viele backup Programme unterstützen die S3 Schnittstelle
- Da S3 ”aus der Cloud” kommt, sind diese Features gängig:
  - Datenverschlüsselung
  - Robustheit gegen Netzwerkfehler

Minio II

```
root@mbk-test:~# ./minio server /data/minio/
Endpoint:  http://172.23.0.102:9000  http://127.0.0.1:9000
AccessKey: 54CC
SecretKey: 4q4X

Browser Access:
   http://172.23.0.102:9000  http://127.0.0.1:9000
```

Danach ein ssh port-forward von localhost:9000 auf den container

```
slogin -l mnbokaem -p 22102 -L 9000:127.0.0.1:9000 storage.franken.de
```

Minio III

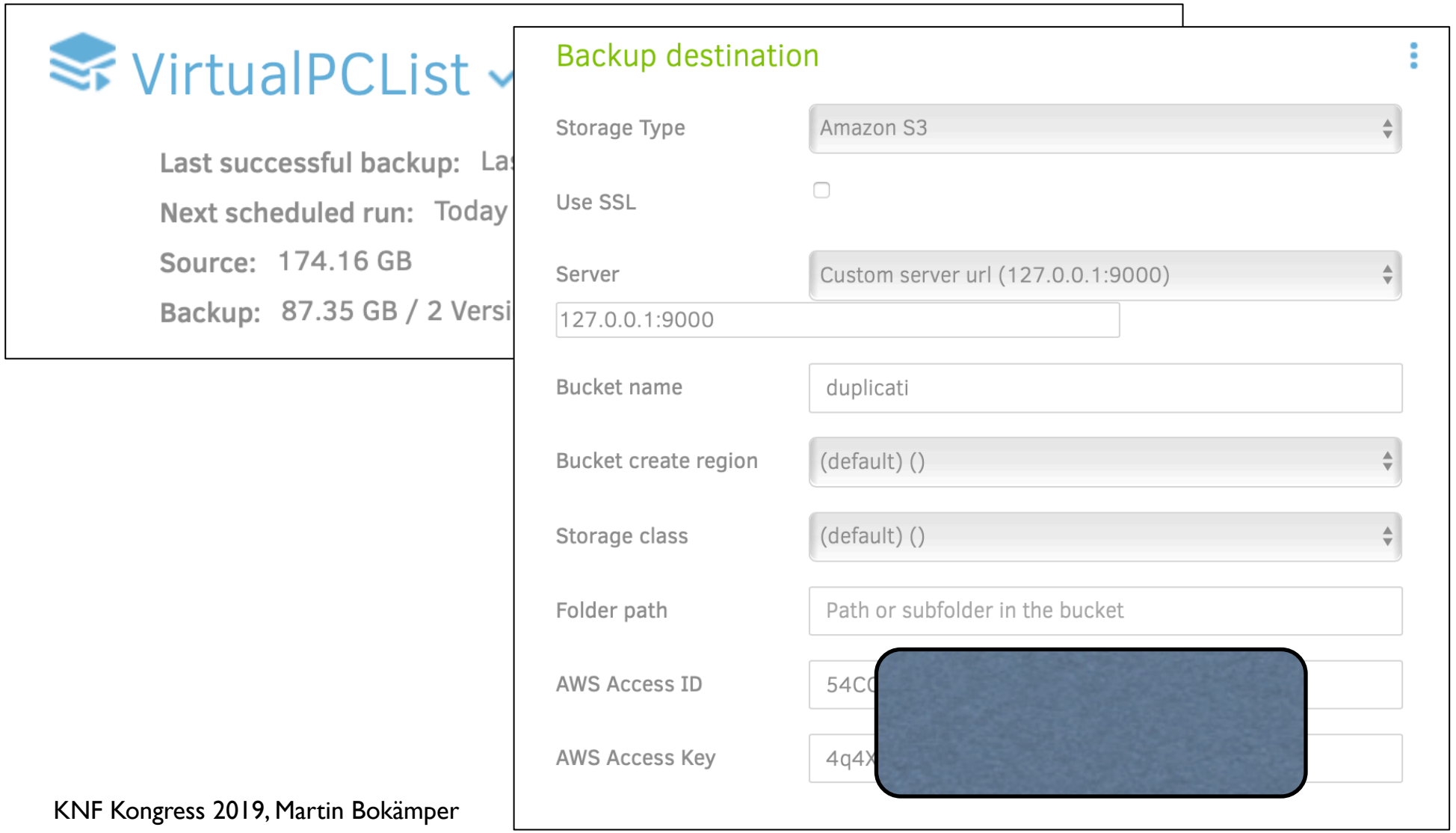
Die Zahl der S3-kompatiblen Programme ist riesig

Von https://github.com/minio/awesome-minio :

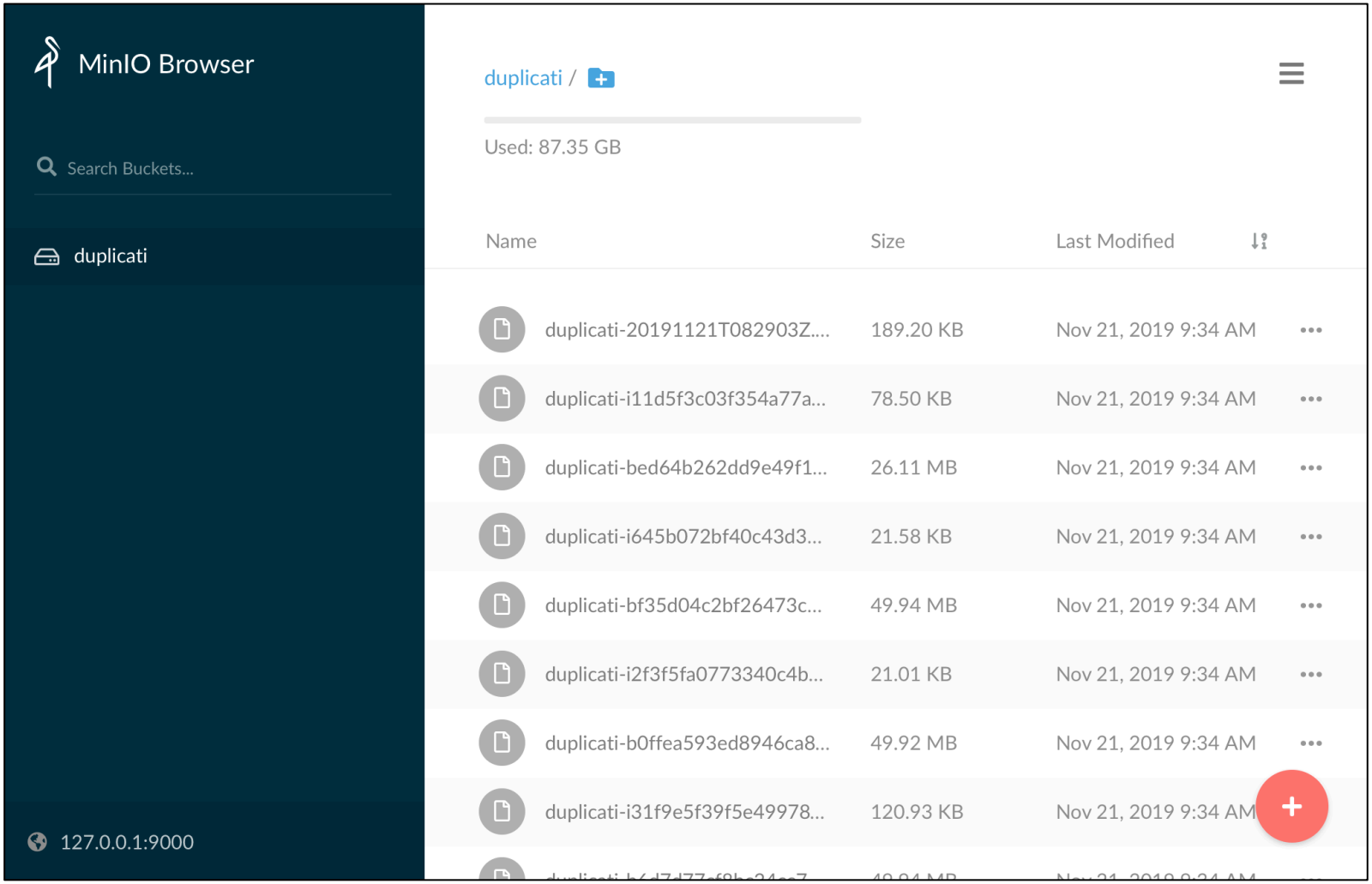
Cloud Backup / Versioning

- Arq - Arq is a storage backend agnostic backup tool for Mac and Windows. Backend services include Amazon Cloud Drive, Google Drive, Dropbox, One Drive, Amazon S3 and more. It also supports 'Other S3-Compatible Services' which means that you can use Minio to build your own backend.
- BackupHive - Providing online backup services from The Netherlands with Minio as an S3 compatible back-end server to store and retrieve files. Minio is very scalable, uses almost no resources itself and is easy to maintain. The awesome team has a strong combined knowledge of use-cases, ranging from the smallest personal project to large scale cross-datacenter setups, all available within a comfortable community.
- burry - Burry, the BackUp & RecoveRY tool for cloud native infrastructure services enables to backup and restore ZooKeeper, etcd and Consul to and from local storage, Amazon S3, Azure Storage, Google Storage, Minio.
- CloudBerry Backup - CloudBerry Backup is used to store files, folders and system images to cloud storage providers. CloudBerry uses Minio for standalone, online and managed backup service.
- Duplicati - Duplicati is free, open source, backup software that implements full encryption, compression, and de-duplication that fully obscures backup contents from data hosting providers. It supports S3-compatible services, allowing Minio to server as the backend storage.
- rclone - "rsync for cloud storage". Rclone is a command line application to sync files to and from cloud storage systems and it works well with Minio. Check out rclone's s3 docs for more information.
- restic - restic is a backup program that is fast, efficient and secure. Check the documentation for instructions on how to backup to a Minio server using restic.
- s3git - git for cloud storage. s3git provides distributed version control for data. Create decentralized and versioned repos that scale infinitely to 100s of millions of files. Clone huge PB-scale repos on your local SSD to make changes, commit and push back. Check out s3git docs for more information.

Minio IV - duplicati



Minio V - duplicati



Spielregeln I

Alle privaten Nutzdaten ~~sollen~~ MÜSSEN verschlüsselt sein

- Der KNF übernimmt keine Verantwortung für deren Sicherheit
- Sollten diese “ausgespäht” werden, darf das keine Katastrophe sein. Nur durch Verschlüsselung sind diese Daten sicher.

Die Container sind für die private Datensicherung & archivierung

- Keine ”Coin-Miner”, “Spamserver”, “Netzscanner” oder ähnliches
- Im Zweifelsfall den Kontakt suchen mit dem Team unter storage@franken.de

Spielregeln II

Ressourcen:

- Der Plattenplatz “ist Deiner”
- RAM und IOPS sind geteilt – bitte vernünftig damit umgehen
  - “Dicke Services” (> 1GB RAM) nicht permanent laufen lassen
  - Konsistenz-Checks nicht zu oft (z.B. einmal pro Monat)

Keiner braucht Backups

Niemand braucht eine “Backup Lösung”

RESTOREs sind das reale Ziel
Backups sind nur das notwendige übel um diese zu ermöglichen

Daher: IMMER mal eine Probe machen:

Kann ich meine Daten tatsächlich wiederbekommen?

- Auch wenn alle aktuellen Rechner weg sind ?
- Auch in 5 Jahren wenn ich alle Details vergessen habe ?

Hier besonders wichtig: Der Key zur Datenentschlüsselung
mehrfach auf Papier und anderen Medien halten

Tester gesucht

- Als Tester nur erfahrene KNF Mitglieder
- Ein public ssh-key für den Zugang
  [potentiell ein separater key für storage]
- Zeit um sich mit dem Thema etwas zu beschäftigen

Anmeldung in der Kaffeepause!

Vielen Dank!

Gibt es noch Fragen ?

Anmeldung für weitere Tester in der Kaffeepause

Links:

- Borg Backup: https://www.borgbackup.org/
- Minio: https://min.io