Journey of a Home-based Personal Cloud Storage Project

SCALE 21x
Julien RIOU
March 16, 2024
2007
Ubuntu Party, Paris

May 2007
Los Angeles

August 2007
17 years later
Who am I?

• Julien RIOU
• Open Source DBA
• https://julien.riou.xyz
• @jriou@hachyderm.io
Summary

1. Why?
2. History
3. Infrastructure
4. Data management
5. Alerting
6. Observability
7. Automation
8. What’s next?
9. Takeaways
Why?

Home-based Personal Cloud Storage, why on earth?
Why?

Home-based Personal Cloud Storage, why on earth?

• Never lose data again
• Control my data
• Learn new stuff
• Have fun!
History
Apartment
2013
USB drives
USB drives

- Hard to find
- NTFS (because Microsoft Windows)
- Physically plug, automount
- Umount/eject, unplug
Network Attached Storage (NAS)
Shared NAS

- Desktop PC
- Home office
- SMB shares with Samba
- Breaking upgrades
New job

2015

- Major cloud provider in Europe
- Discount price on HDDs (not anymore)
- OpenZFS (NFS, CIFS)
- GNU/Linux on servers and desktops
Small storage

- Must be small and silent
- Synology design
- 3x4TB HDD at discount price
- Intel NUC motherboard, PCI RAID card
- FreeBSD for built-in OpenZFS support
Motherboard sizes

MINI ITX

MICRO ATX
Copying data…
Big storage

- Classic ATX tower
- 3x2TB HDD at discount price
- FreeBSD
Baby
2018

- Put the computers away to the basement
- Time better spent with my baby
New house

2019

• More space!
• Noise is not an issue anymore
• Secure basement
Old storage

- Rebuilt my main computer
- Re-used my old computer as a storage server
  - The first computer I’ve ever built in 2008
- 3x1TB HDD from my stock
Issues

- USB stick not bootable
- CD-ROM of FreeBSD 12 had a LUA error
  - FreeBSD 11 too
  - Debian 10 worked
- Freezes
  - Hard reboot
- Fully replaced and upgraded today (3x2TB)
## Recap

2024

<table>
<thead>
<tr>
<th>Description</th>
<th>Name</th>
<th>Capacity (TiB)</th>
<th>OS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Big storage</td>
<td>storage1</td>
<td>5.45</td>
<td>Debian</td>
</tr>
<tr>
<td>Old storage</td>
<td>storage2</td>
<td>5.45</td>
<td>Debian</td>
</tr>
<tr>
<td>Small storage</td>
<td>storage3</td>
<td>10.9</td>
<td>Debian</td>
</tr>
</tbody>
</table>
COMPUTERS

COMPUTERS EVERYWHERE
Infrastructure
Clients
Operating systems

- No more Microsoft Windows
- Ubuntu and friends
Network File System (NFS)

- Easy to set up
- Easy to maintain
- Mount a remote directory locally
- Harder to install and maintain
  - Easier with Docker
- User friendly
  - Drive client, Web UI (seahub)
- Keep files in sync
  - Pinned full files, full files and placeholders
Connectivity
Static IP address
Static IP address

Fixed IP Option

€30.25 / month

Disposez d'une adresse IP Fixe pour accéder à distance à votre serveur au moyen d'une connexion internet.

Read less ▲
Static IP address

Fixed IP Option

€30.25 / month

Disposz d'une adresse IP Fixe pour accéder à votre serveur au moyen d'une connexion internet.

Read less ▲
ISP modem settings
ISP modem settings

- SSH, HTTP and HTTPS closed by default
- Port mapping
- Request the ISP to set security level to low
- It worked at the apartment, not in the house
ISP modem settings

- SSH, HTTP and HTTPS closed by default
- Port mapping
- Request the ISP to set security level to low
- It worked at the apartment, not in the house
• Virtual Private Network (VPN)
• Client-server model
• Authentication with certificates
• TLS
• Client-to-client allowed
• Static IP address assignment to clients
Custom settings

topology subnet
server 10.xx.xx.xx 255.xx.xx.xx
client-to-client
client-config-dir /etc/openvpn/ccd
ifconfig-pool-persist /var/log/openvpn/ipp.txt

; declare a subnet like home
; with the range you like
; allow clients to talk to each other
; static IP configuration per client
; IP lease settings
Remote administration

- Secure Shell protocol (SSH)
- Login and execute commands on a remote host
Data management
Disk management

- Zettabyte File System (ZFS)
- Volume manager, RAID-Z
- Filesystems
- Snapshots
  - Performance!
  - Replication, cloning, rollback
- Compression, encryption
- Production ready, even on Linux
RAID-Z

storage1 ~ # zpool status
  pool: storage
  state: ONLINE
    scan: scrub repaired 0B in 02:59:40 with 0 errors on Sun Feb 11 03:23:41 2024
config:

<table>
<thead>
<tr>
<th>NAME</th>
<th>STATE</th>
<th>READ</th>
<th>WRITE</th>
<th>CKSUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>storage</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>raidz1-0</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>sda</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>sdb</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>sdc</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

errors: No known data errors
RAID-Z

storage1 ~ # zpool list

<table>
<thead>
<tr>
<th>NAME</th>
<th>SIZE</th>
<th>ALLOC</th>
<th>FREE</th>
<th>CKPOINT</th>
<th>EXPANDSZ</th>
<th>FRAG</th>
<th>CAP</th>
<th>DEDUP</th>
<th>HEALTH</th>
<th>ALTROOT</th>
</tr>
</thead>
<tbody>
<tr>
<td>storage</td>
<td>5.44T</td>
<td>2.89T</td>
<td>2.55T</td>
<td>-</td>
<td>-</td>
<td>8%</td>
<td>53%</td>
<td>1.00x</td>
<td>ONLINE</td>
<td>-</td>
</tr>
</tbody>
</table>
## Compression

```
storage1 ~ # zfs get compression storage
NAME     PROPERTY     VALUE     SOURCE
storage  compression  lz4       local
```
Filesystems

```
storage1 ~ # zfs list -t filesystem
NAME        USED  AVAIL     REFER  MOUNTPOINT
storage     1.93T  1.58T      139K  /storage
storage/julien  348G  1.58T      338G  /storage/julien
```
Snapshots

```
storage1 ~ # zfs list -t snapshot -r storage/julien | tail -n 3
storage/julien@autosnap_2024-02-25_00:00:01_daily       0B      -      338G  -
storage/julien@autosnap_2024-02-26_00:00:02_daily       0B      -      338G  -
storage/julien@autosnap_2024-02-27_00:00:02_daily       0B      -      338G  -
```
Replication

```
zfs send POOL/FS@SNAPSHOT-1 | ssh REMOTE_HOST zfs recv POOL/FS
zfs send -i POOL/FS@SNAPSHOT-1 POOL/FS@SNAPSHOT-2 | ssh REMOTE_HOST zfs recv POOL/FS
zfs send -i POOL/FS@SNAPSHOT-2 POOL/FS@SNAPSHOT-3 | ssh REMOTE_HOST zfs recv POOL/FS
```
Snapshot management

Policy-driven snapshot management tool for ZFS filesystems

- Take snapshots
- Pre and post snapshot scripts
- Prune snapshots
- Monitoring (health, capacity)
Templates configuration

[template_main]
hourly = 0
daily = 31
monthly = 12
yearly = 10
autosnap = yes
autoprune = yes

[template_archive]
hourly = 0
daily = 31
monthly = 12
yearly = 10
autosnap = no
autoprune = yes
Policies

[storage/julien]
  use_template = main

[storage/dad]
  use_template = archive
Job definition
systemd service

```
storage1 ~ # systemctl cat sanoid.service
# /lib/systemd/system/sanoid.service
[Unit]
Description=Snapshot ZFS filesystems
Documentation=man:sanoid(8)
Requires=local-fs.target
After=local-fs.target
Before=sanoid-prune.service
Wants=sanoid-prune.service
ConditionFileNotEmpty=/etc/sanoid/sanoid.conf

[Service]
Type=oneshot
Environment=TZ=UTC
ExecStart=/usr/sbin/sanoid --take-snapsshots --verbose
```
Job scheduling

systemd timer

storage1 ~ # systemctl cat sanoid.timer
# /lib/systemd/system/sanoid.timer
[Unit]
Description=Run Sanoid Every 15 Minutes

[Timer]
OnCalendar=*:0/15
Persistent=true

[Install]
WantedBy=timers.target
Job scheduling
systemd timer

storage1 ~ # systemctl cat sanoid.timer
# /lib/systemd/system/sanoid.timer

[Unit]
Description=Run Sanoid Every 15 Minutes

[Timer]
OnCalendar=*:0/15
Persistent=true

[Install]
WantedBy=timers.target

storage1 ~ # systemctl list-timers sanoid.timer --all

NEXT LEFT LAST PASSED UNIT ACTIVATES
Tue 2024-02-27 09:00:00 CET 11min left Tue 2024-02-27 08:45:01 CET 3min 20s ago sanoid.timer sanoid.ser

1 timers listed.
Snapshot replication

- **Syncoid**
  - included with Sanoid
- **rsync**-like
- Resume on interruption
- Bandwidth control
Usage

```
/usr/sbin/syncoid                       
   storage/julien                      
   zfs@REMOTE_STORAGE:storage/julien   
   --no-sync-snap                      
   --source-bwlimit=512k
```
Usage

```
/usr/sbin/syncoid  
storage/julien  
zfs@REMOTE_STORAGE:storage/julien  
--no-sync-snap  
--source-bwlimit=512k
```

Added to `/opt/syncoid.sh` script
Job definition

systemd service

```bash
storage1 ~ # systemctl cat syncoid.service
#/etc/systemd/system/syncoid.service
[Unit]
Description=Send ZFS snapshots created by Sanoid
Requires=zfs.target
After=zfs.target

[Service]
Type=oneshot
User=zfs
ExecStart=-/opt/syncoid.sh

[Install]
WantedBy=multi-user.target
```
Job scheduling

systemd timer

storage1 ~ # systemctl cat syncoid.timer
#/etc/systemd/system/syncoid.timer
[Unit]
Description=Run Syncoid every night

[Timer]
OnCalendar=**-** 00,04:30:00 UTC
AccuracySec=1us
RandomizedDelaySec=2h30

[Install]
WantedBy=timers.target
Client replication
Replication overview
Health

```
storagel ~ # sanoid --monitor-snapshots
OK: all monitored datasets (storage/dad, storage/julien) have fresh snapshots
```
Health

storage1 ~ # sanoid --monitor-snapshots
OK: all monitored datasets (storage/dad, storage/julien) have fresh snapshots

storage1 ~ # sanoid --monitor-health
OK ZPOOL storage : ONLINE {Size:5.44T Free:2.55T Cap:53%}
Alerting
Nagios

- Nagios Core
- Simple configuration files
- Web UI
- Plugins
Welcome to pilote!

- Runs on Raspberry Pi
- Debian based distribution
Components

- Hosts
- Hostgroups
- Services
- Notifications
Host

/etc/nagios4/conf.d/hosts.cfg

define host {
    use home-host
    host_name storage1
    alias storage1
    address 169.254.0.1
}
Hostgroups

/etc/nagios4/conf.d/hostgroups.cfg

define hostgroup {
    hostgroup_name  storage-servers
    alias           Storage servers
    members         storage1,storage2,storage3
}

Services commands

- check_ping
- check_nrpe
  - Nagios Remote Plugin Executor
- check_http
Services states

- OK
- WARNING
- CRITICAL
- UNKNOWN
define service {
  use home-service
  hostgroup_name storage-servers
  service_description zfs_snapshots
  check_command check_nrpe!check_zfs_snapshots
}

Service configuration
NRPE agent

/etc/nagios/nrpe_local.cfg

command[check_zfs_snapshots]=/usr/bin/sudo /usr/sbin/sanoid --monitor-snapshots
Notifications

Send Nagios notifications to a Telegram Messenger channel.

notify-by-telegram
***** Nagios *****
Notification Type: PROBLEM
Service: zfs_snapshots
Host: storage3
Address: 
State: CRITICAL
Date/Time: Sun Dec 24 09:04:35 CET 2023
Additional Info:
CRIT: storage/[redacted]'s newest daily snapshot is 1d 8h 4m 33s old (should be 1d 8h 0m 0s), CRIT: storage/[redacted]'s newest daily snapshot is 1d 8h 4m 33s old (should be 1d 8h 0m 0s), CRIT: storage/[redacted]'s newest daily snapshot is 1d 8h 4m 34s old (should be 1d 8h 0m 0s)
***** Nagios *****
Notification Type: RECOVERY
Service: zfs_health
Host: storage2
Address: 
State: OK
Date/Time: Wed Jan 17 21:47:04 CET 2024
Additional Info:
OK ZPOOL storage : ONLINE
{Size:5.45T Free:2.59T Cap:52%}
Web UI
External access

HTTPS
nginx

HTTP
nagios

pilote

VPN
### Host Status Totals

<table>
<thead>
<tr>
<th>Status</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Up</td>
<td>14</td>
</tr>
<tr>
<td>Down</td>
<td>0</td>
</tr>
<tr>
<td>Unreachable</td>
<td>0</td>
</tr>
<tr>
<td>Pending</td>
<td>0</td>
</tr>
</tbody>
</table>

### Service Status Totals

<table>
<thead>
<tr>
<th>Severity</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>OK</td>
<td>157</td>
</tr>
<tr>
<td>Warning</td>
<td>0</td>
</tr>
<tr>
<td>Unknown</td>
<td>0</td>
</tr>
<tr>
<td>Critical</td>
<td>0</td>
</tr>
<tr>
<td>Pending</td>
<td>0</td>
</tr>
</tbody>
</table>

### Host Status Details

- **14** Up services:
  - apei_battery_charge: OK
  - apei_load: OK
  - apei_status: OK
  - apei_tank_left: OK
  - backha_str: OK
  - backha_str: OK
  - disk_rsc: OK
  - load: OK
  - ntp: OK
  - opensim: OK
  - opensim_cmd: OK
  - sched_maint: OK
  - tank_gps: OK
  - tank_gps: OK
  - users: OK
  - zfs_capacity: OK
  - zfs_health: OK
  - zfs_snapshots: OK
  - zpool_progs: OK

- **0** Down services

- **0** Unreachable services

- **0** Pending services

### Service Information

- OK - Battery Charge: 100.0%
- OK - Load: 13.0%
- OK - Power Line: ONLINE
- PROCOK 1 process with command name 'backha_str'
- PROCOK 1 process with command name 'backha_str'
- DISK OK - free space / 16749468 (95% used=50%)
- LOAD OK - scaled load average: 0.00, 0.00, 0.00 - total load average: 0.00, 0.00, 0.00
- NTP is healthy
- PROCOK 1 process with command name 'opensim'
- SSL_CERT OK - localhost:443, https, x509 certificate 'storage' from Easy-RSA CA valid until May 3 06:07:24 2023 GMT (expires in 831 days)
- PROCOK 2 processes with args 'corosync'
- PROCOK 1 process with command name 'tank_gps'
- PROCOK 1 process with command name 'zfs_capacity'
- OK ZPOOL storage: 52%
- OK ZPOOL storage: ONLINE (Size 5.44T Free 2.57T Cap 52%)
- OK all monitored datasets (storage) have fresh snapshots
- PROCOK 0 processes with 51816 "z"
Observability

- Disk space evolution
- Network stability
- Elephant
- Temperature in the room
- Power consumption
TIG stack

- Telegraf
- InfluxDB
- Grafana
telegraf

The plugin-driven server agent for collecting & reporting metrics.

https://github.com/influxdata/telegraf
Inputs

[[inputs.cpu]]
  percpu = false
  totalcpu = true
  collect_cpu_time = false
  report_active = false

[[inputs.diskio]]
  devices = ['sda', 'sdb', 'sdc', 'sdd']
Outputs

[[outputs.influxdb]]
urls = ["https://x.x.x.x:8088"]
database = "metrics"
skip_database_creation = true
username = "telegraf"
password = "****"
insecure_skip_verify = true
content_encoding = "gzip"
Scalable datastore for metrics, events and real-time analytics

https://github.com/influxdata/influxdb
The open-source platform for monitoring and observability

https://github.com/grafana/grafana
Grafana dashboard

Storage

Allocated

Capacity

Fragmentation

Disk temperatures

Error rates

Disks uptime

sda 8.01 year
sdb 8.32 year
sdc 4.60 year
sdd 1.40 year
Overview

Diagram showing the components: HTTPS, nginx, VPS, storage1, storage2, storage3, telegraf, InfluxDB, Grafana, metrics server, VPN.
Notes

- Docker images available
  - InfluxDB, Grafana
- Not MicroSD friendly, prefer SSD or HDD
- Consider using Prometheus for the future
Sensors

- Temperature
- Humidity
- Noise
Hardware

- Arduino Uno (Elegoo Uno R3)
  - Powered by USB
- DHT22 sensor (temperature, humidity)
- KY-037 sensor (sound)
- Breadboard
- Cables
Software

• Arduino IDE
• Upload sketch to the board
Sketch
Definitions

#include <DHT.h>
#define KYPIN A0  // analog pin where KY-037 sensor is connected
#define DHTPIN 2  // digital pin where DHT22 sensor is connected

DHT dht(DHTPIN, DHT22); // initialize DHT22 object

float h;  // humidity
float t;  // temperature
int s;    // sound
Setup

```cpp
void setup()
{
    Serial.begin(9600);
    dht.begin();
}
```
void loop()
{
    // sensors need some time to produce valid values
    delay(2000);

    // read values from sensors
    h = dht.readHumidity();
    t = dht.readTemperature();
    s = analogRead(KYPIN);
Main loop (2/2)

```cpp
// print "<humidity>,<temperature>,<sound>" (CSV-like)
if (isnan(h) && !isnan(t) && !isnan(s)) {
    Serial.print(h);
    Serial.print(",");
    Serial.print(t);
    Serial.print(",");
    Serial.println(s);
}
```
Multiplexing

- Serial port can be accessed by only one program
- MQTT Broker (Mosquitto)
- serial2mqtt
- Nagios check-mqtt
- Telegraf mqtt_consumer
How is the temperature?
Humidity
Noise
Noise
Noise
Power consumption

How much will it cost?
Average monthly electricity wholesale price in Belgium from January 2019 to January 2024 (in euros per megawatt-hour)
Household electricity prices worldwide in June 2023, by select country (in U.S. dollars per kilowatt-hour)
Wattmeter
Uninterruptible power supply (UPS)

- Apcupsd (APC UPS daemon) compatible
- Telegraf plugin
- Grafana dashboard already available
- Save from power outage
- A little bit pricey (€164,23 in 2020)
Yearly cost

$7/y
In real life
Automation
Failures happen
Failures happen

MicroSD cards with I/O errors
Failures happen

Flood or fire in the house
Failures happen

WELL

THAT ESCALATED QUICKLY
Deployments
Deployments

1. Install the operating system
2. Install and configure software
3. Restore data (optional)
Deployments

1. Install the operating system
2. Install and configure software
3. Restore data (optional)
Ansible is a radically simple IT automation system.

https://github.com/ansible/ansible
Concepts

- **Inventory**: combination of
  - **Hosts**: remote machine to manage
  - **Groups**: hosts sharing a common attribute
- **Playbook**: list of tasks executed in order, on groups
  - **Roles**: group of tasks that can be shared to the world
  - **Tasks**: module + arguments
    - **Modules**: smallest unit of code to execute on hosts
Inventory

inventory/hosts file

[all]
vps ansible_host=xxx.xxx.xxx.xxx
pilote ansible_host=xxx.xxx.xxx.xxx
metrics ansible_host=xxx.xxx.xxx.xxx
storage1 ansible_host=xxx.xxx.xxx.xxx
storage2 ansible_host=xxx.xxx.xxx.xxx
storage3 ansible_host=xxx.xxx.xxx.xxx

[storage]
storage1 ansible_host=xxx.xxx.xxx.xxx
storage2 ansible_host=xxx.xxx.xxx.xxx
storage3 ansible_host=xxx.xxx.xxx.xxx
Playbook overview

site.yml

- import_playbook: common.yml
- import_playbook: storage.yml
- import_playbook: ...
Playbook overview

site.yml

- import_playbook: common.yml
- import_playbook: storage.yml
- import_playbook: ...

common.yml

- hosts: all
  roles:
  - common
storage.yml

- hosts: storage
  roles:
    - zfs
    - openvpn
    - sanoid
    - ...

- hosts: storage1
  roles:
    - nfs
Role example

roles/sanoid/
|-- defaults
|   |-- main.yml
|-- handlers
|   |-- main.yml
|-- tasks
|   |-- main.yml
|-- templates
|   |-- sanoid.conf.j2
|   |-- syncoid.service.j2
|   |-- syncoid.sh.j2
|   |-- syncoid.timer.j2
Module examples

- ansible.builtin.apt
- ansible.builtin.file
- ansible.builtin.service
- ansible.builtin.template
Template example

Task

- name: Deploy Syncoid script
  ansible.builtin.template:
    src: syncoid.sh.j2
    dest: /opt/syncoid.sh
    owner: zfs
    group: root
    mode: "0750"
#!/bin/bash
{{ ansible_managed | comment }}

{% for dataset in sanoid_main_datasets %}{% for destination in syncoid_destinations %}
echo "Sending {{ dataset }} to {{ destination }}"
/usr/sbin/syncoid {{ dataset }} {{ syncoid_user }}@{{ destination }}:{{ dataset }} \
   --no-sync-snap \n   {% if syncoid_source_bwlimit %}--source-bwlimit={{ syncoid_source_bwlimit }} {% endif %}
{% endfor %}{% endfor %}
{% endfor %}{% endfor %}
Result on the managed host

```bash
#!/bin/bash
#
# Ansible managed
#

echo "Sending storage/julien to xxx.xxx.xxx.xxx"
/usr/sbin/syncoid storage/julien xxx@xxx.xxx.xxx.xxx:storage/julien \
    --no-sync-snap \  
    --source-bwlimit=512k
echo "Sending storage/dad to xxx.xxx.xxx.xxx"
/usr/sbin/syncoid storage/dad xxx@xxx.xxx.xxx.xxx:storage/dad \
    --no-sync-snap \  
    --source-bwlimit=512k
```
Upgrades
Upgrades

upgrade.yml

- name: Upgrade systems
  hosts: all
  tasks:
    - include_tasks: tasks/apt-upgrade.yml
Upgrades

upgrade.yml

- name: Upgrade systems
  hosts: all
  tasks:
    - include_tasks: tasks/apt-upgrade.yml

tasks/apt-upgrade.yml

- name: Run apt upgrade
  ansible.builtin.apt:
    update_cache: true
    upgrade: dist
CLI

ansible-playbook site.yml
ansible-playbook upgrade.yml
What’s next?

- Open-source my Ansible code base
- Automate certificates management
- Use ZFS encryption
- Use Prometheus for metrics
- Forward logs
- Handle mobile phones
Takeaways

- Self-hosting is not that hard
- Consider using TrueNAS
- FOSS is awesome!
- Enjoy what you are doing
Thank you 🙏
Questions
// reveal.js plugins