# 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?

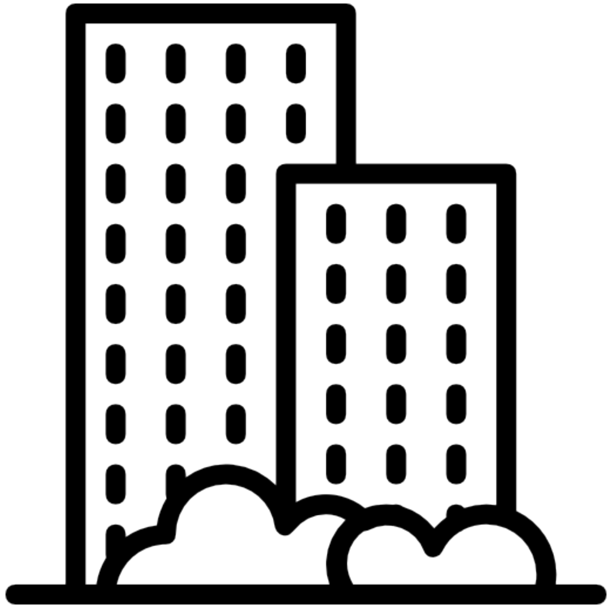
Home-based Personal Cloud Storage, why on earth?

- Never lose data again
- Control my data
- Learn new stuff
- Have fun!

# History

# Apartment

2013



# USB drives

Externe harde schijf 500 GB
Disque dur externe de 500 Go
MEDION
500 GB
500 Go
Plug & Play
Plug & Play
MEDION
Drive-n-go
MEDION
Drive-n-go

# 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

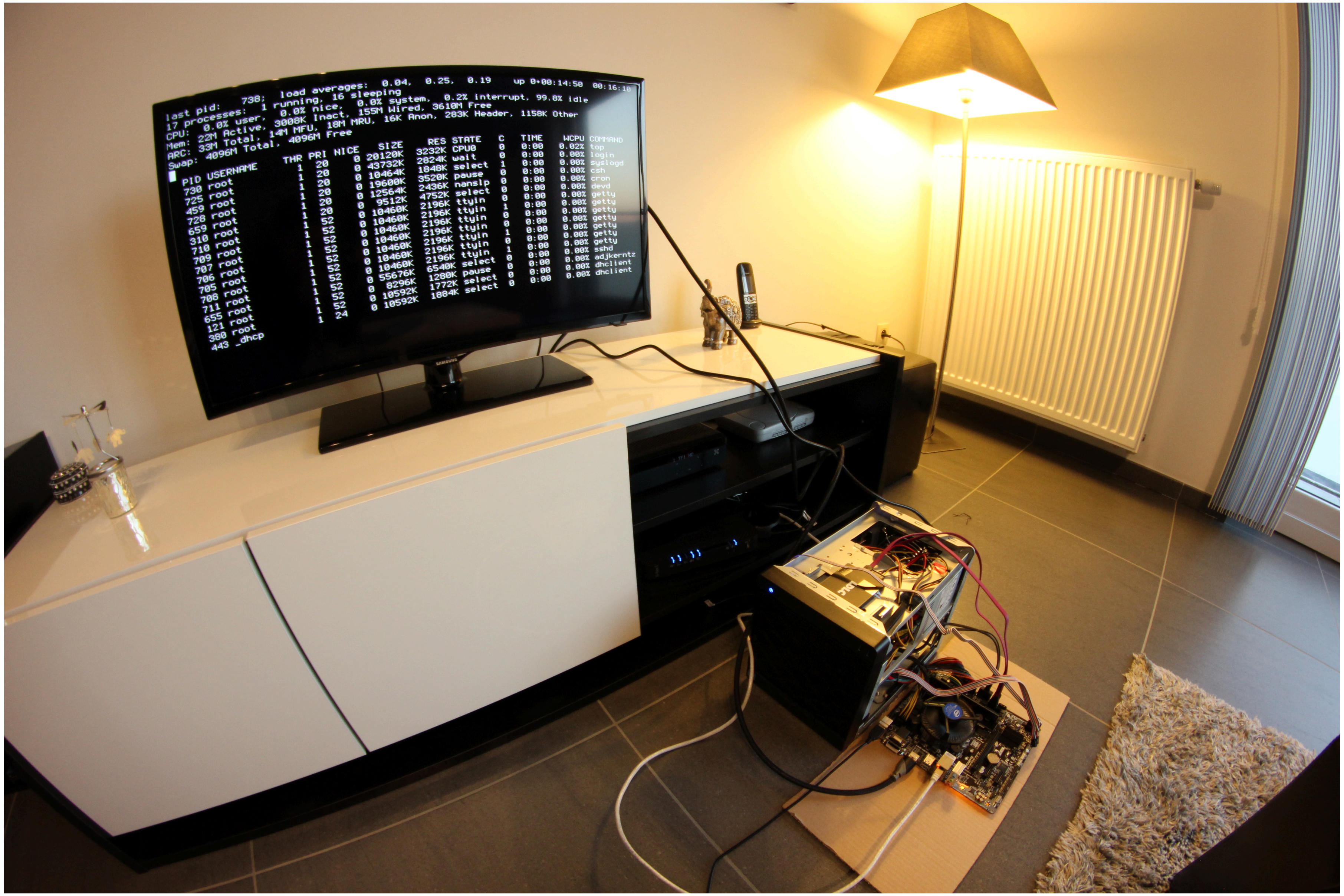
2015

- 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



```
last pid:   738;  load averages:  0.04,  0.25,  0.19    up 0+00:14:50  00:16:10
17 processes:  1 running, 16 sleeping
CPU:  0.0% user,  0.0% nice,  0.0% system,  0.2% interrupt, 99.8% idle
Mem: 22M Active, 3008K Inact, 155M Wired, 3610M Free
ARC: 33M Total, 14M MFU, 18M MRU, 16K Anon, 283K Header, 1158K Other
Swap: 4096M Total, 4096M Free

  PID USERNAME      THR PRI NICE   SIZE    RES STATE   C   TIME    WCPU COMMAND
  730 root            1  20    0 20120K  3232K CPU0    0   0:00   0.02% top
  725 root            1  20    0 43732K  2824K wait    0   0:00   0.00% login
  459 root            1  20    0 10464K  1848K select  1   0:00   0.00% syslogd
  728 root            1  20    0 19600K  3520K pause   0   0:00   0.00% csh
  659 root            1  20    0 12564K  2436K nanslp  0   0:00   0.00% cron
  310 root            1  20    0  9512K  4752K select  0   0:00   0.00% devd
  710 root            1  52    0 10460K  2196K ttyin   0   0:00   0.00% getty
  709 root            1  52    0 10460K  2196K ttyin   1   0:00   0.00% getty
  707 root            1  52    0 10460K  2196K ttyin   0   0:00   0.00% getty
  706 root            1  52    0 10460K  2196K ttyin   0   0:00   0.00% getty
  705 root            1  52    0 10460K  2196K ttyin   1   0:00   0.00% getty
  708 root            1  52    0 10460K  2196K ttyin   0   0:00   0.00% getty
  711 root            1  52    0 10460K  2196K ttyin   1   0:00   0.00% getty
  655 root            1  52    0 55676K  6540K select  0   0:00   0.00% sshd
  121 root            1  52    0  8296K  1280K pause   0   0:00   0.00% adjkerntz
  380 root            1  52    0 10592K  1772K select  0   0:00   0.00% dhclient
  443 _dhcp           1  24    0 10592K  1884K select  0   0:00   0.00% dhclient
```

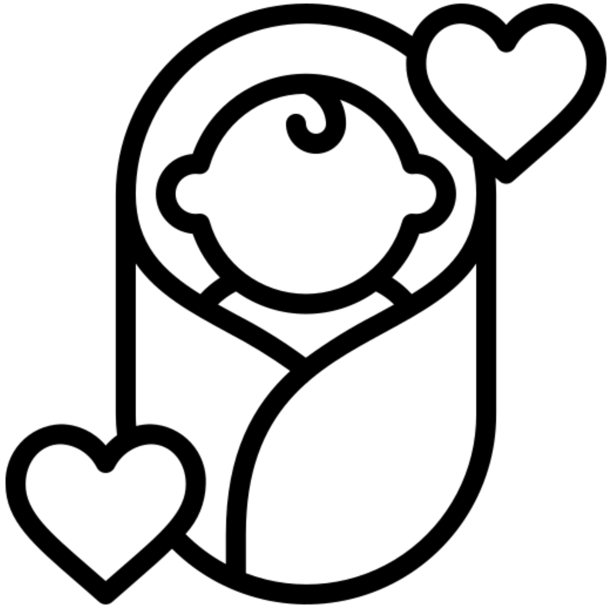
# Copying data…

# 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

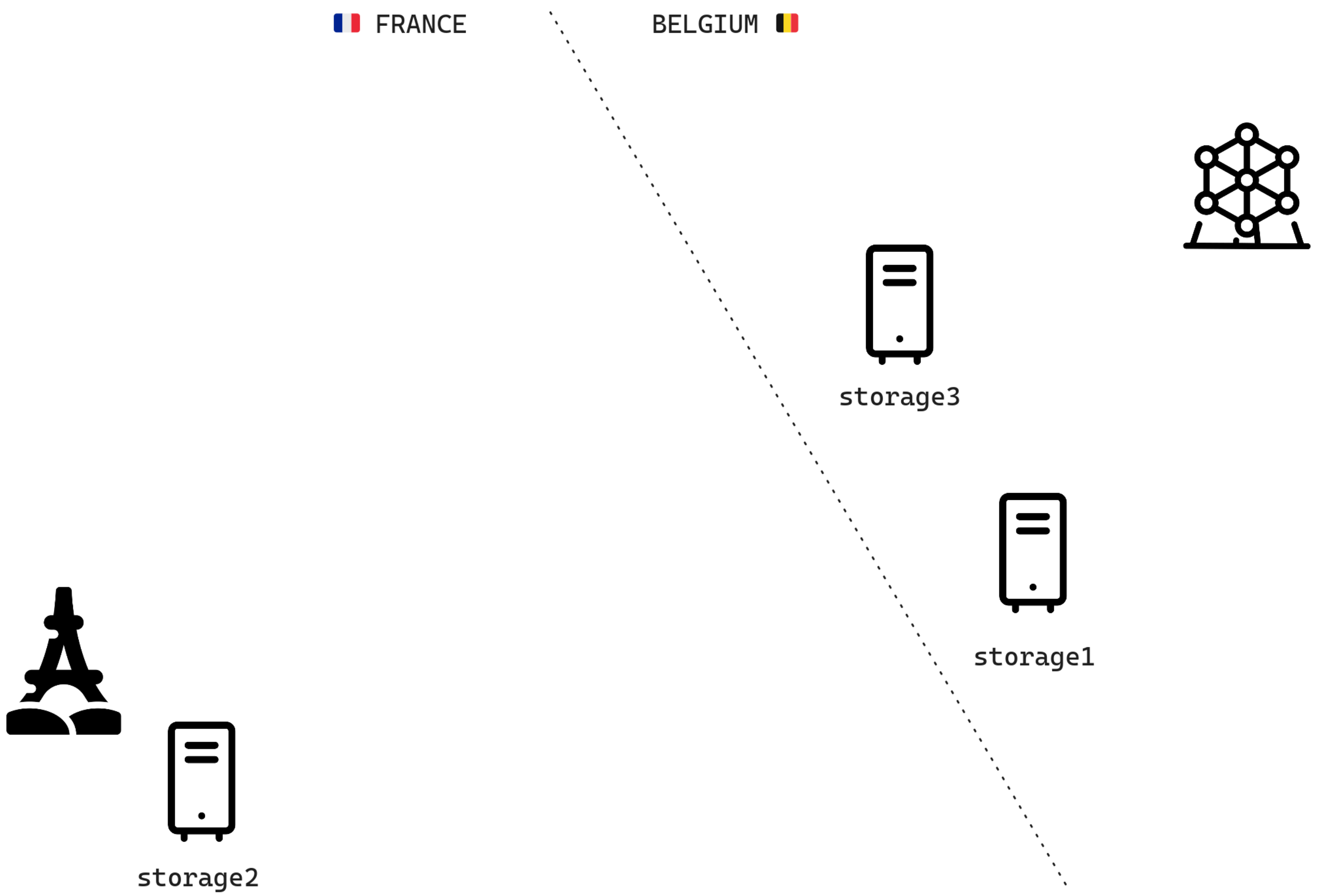
2024

| Description | Name | Capacity (TiB) | OS |
|---|---|---:|---|
| Big storage | storage1 | 5.45 | Debian |
| Old storage | storage2 | 5.45 | Debian |
| Small storage | storage3 | 10.9 | Debian |

COMPUTERS
COMPUTERS EVERYWHERE
SPACE RANGER
LIGHTYEAR
imgflip.com

# Infrastructure

# Map



🇫🇷 FRANCE

BELGIUM 🇧🇪

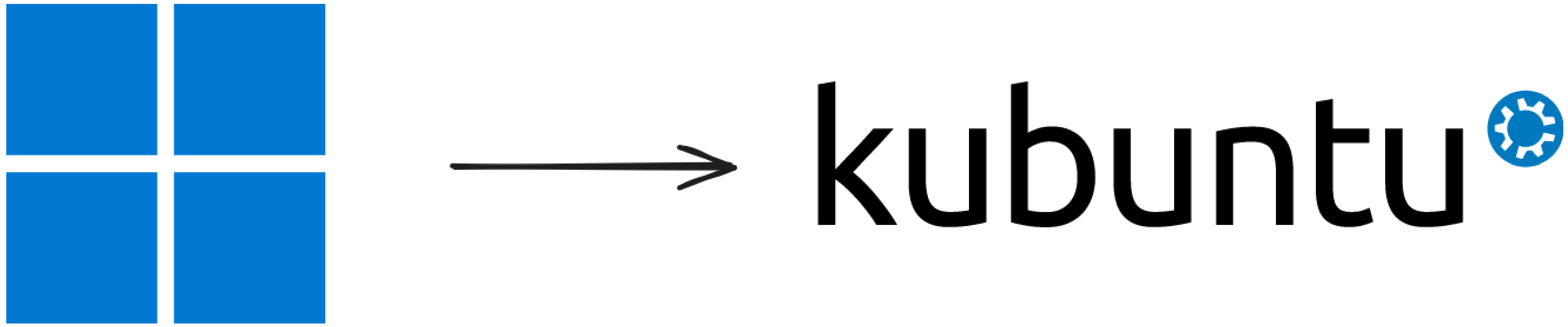
storage3

storage1

storage2

# 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

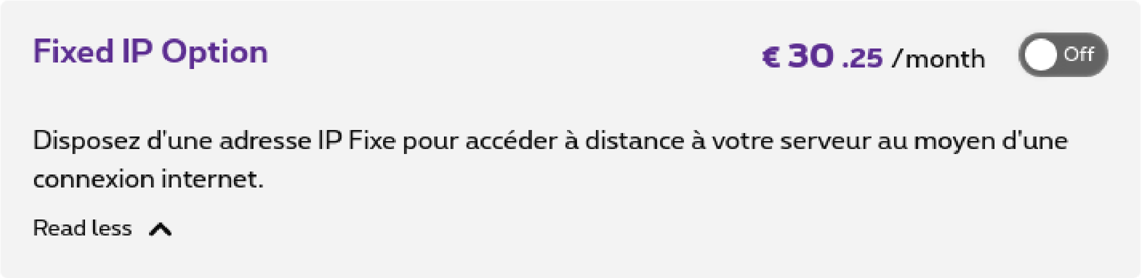
# Seafile

- 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

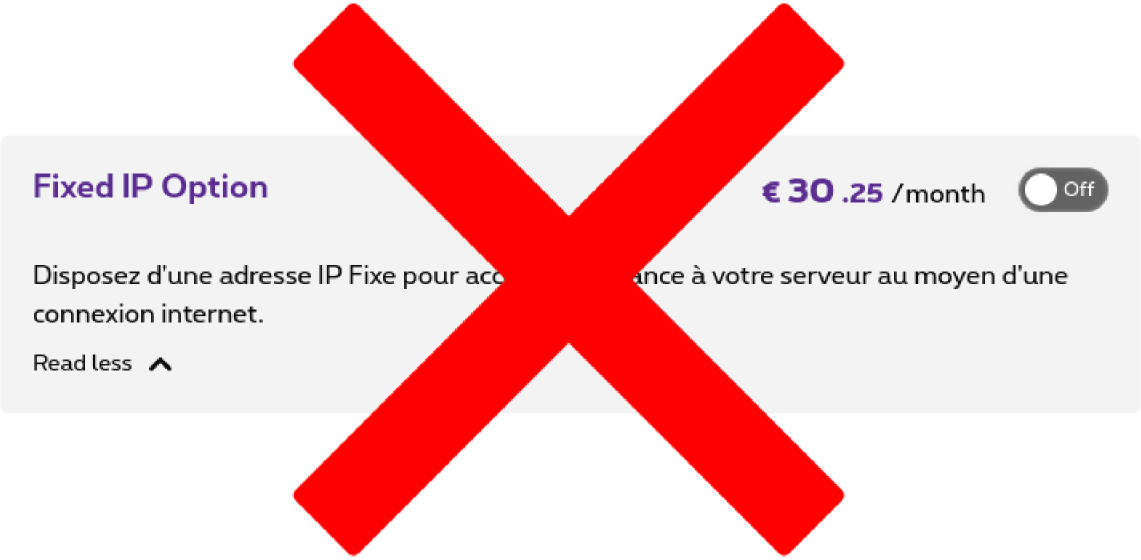


Fixed IP Option

€ 30 .25 /month

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

# Static IP address



Fixed IP Option

€ 30 .25 /month

Off

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

# 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

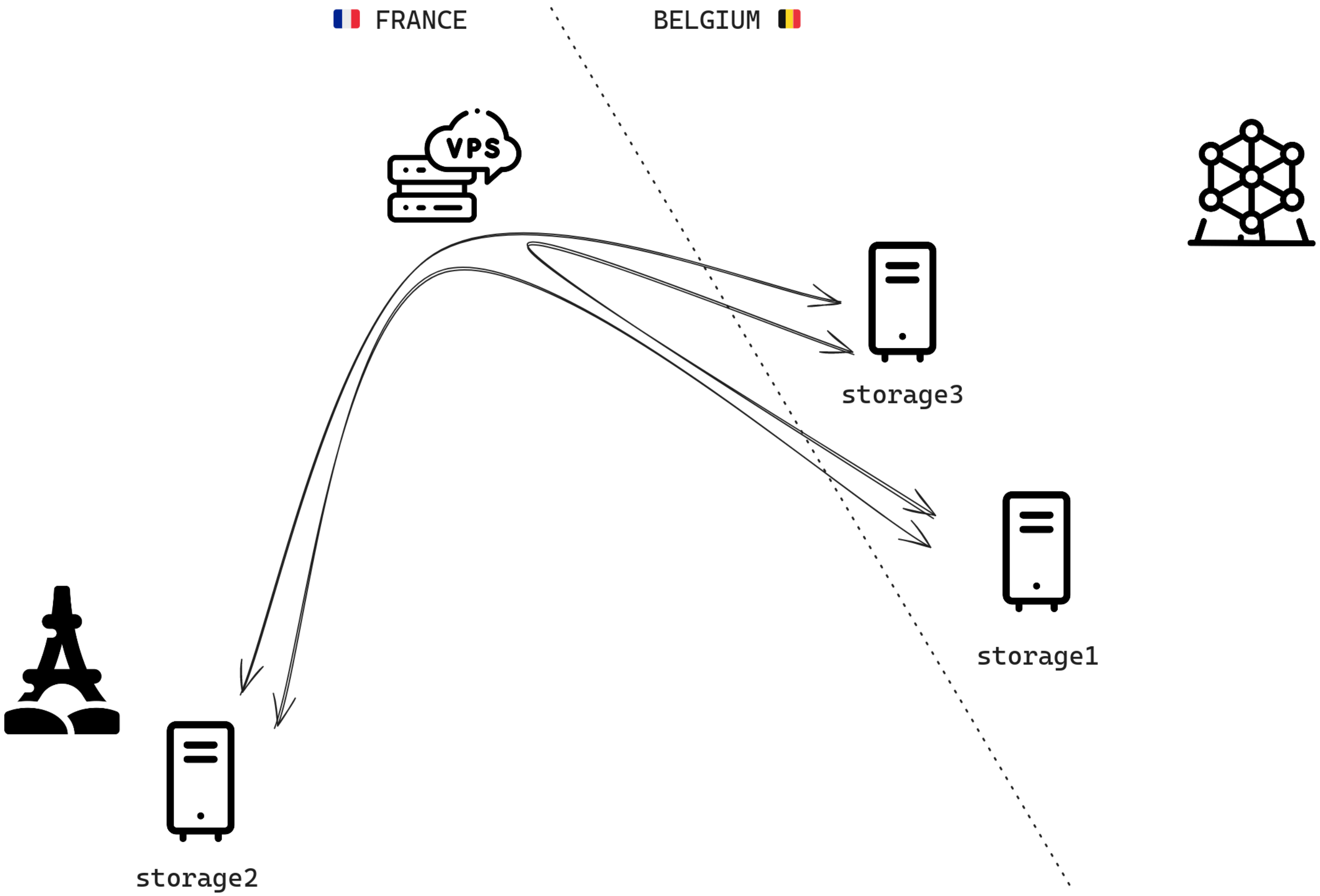
- SSH, HTTP a[illegible] HTTP[illegible] [illegible]osed by default
- Port mapping
- Request the ISP t[illegible] [illegible] **[illegible]curity level** to **low**
- It worked at t[illegible] apartme[illegible] not in the house

# OPENVPN

- 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                                 ; declare a subnet like home
server 10.xx.xx.xx 255.xx.xx.xx                 ; with the range you like
client-to-client                                ; allow clients to talk to each other
client-config-dir /etc/openvpn/ccd              ; static IP configuration per client
ifconfig-pool-persist /var/log/openvpn/ipp.txt  ; IP lease settings
```


🇫🇷 FRANCE
BELGIUM 🇧🇪
VPS
storage3
storage1
storage2

# 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:

        NAME          STATE     READ WRITE CKSUM
        storage       ONLINE       0     0     0
          raidz1-0    ONLINE       0     0     0
            sda       ONLINE       0     0     0
            sdb       ONLINE       0     0     0
            sdc       ONLINE       0     0     0

errors: No known data errors
```

# RAID-Z

```
storage1 ~ # zpool list
NAME      SIZE  ALLOC   FREE  CKPOINT  EXPANDSZ   FRAG    CAP  DEDUP    HEALTH  ALTROOT
storage  5.44T  2.89T  2.55T        -         -     8%    53%  1.00x    ONLINE  -
```

# 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

SANOID

> *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-snapshots --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

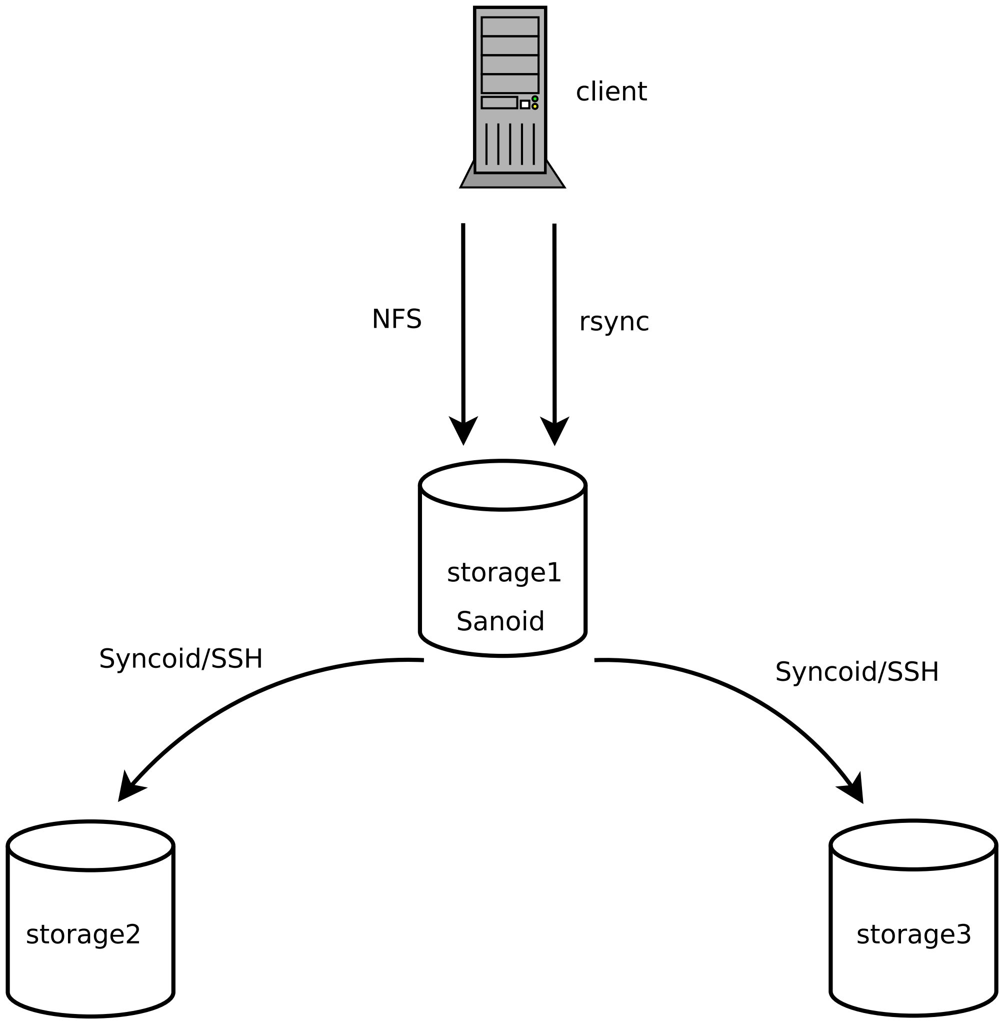
```
storage1 ~ # systemctl cat syncoid.service
# /etc/systemd/system/syncoid.service
[Unit]
Description=Send ZFS snapshots created by Sanoid
Requires=zfs.target
After=zfs.target

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

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

# Job scheduling

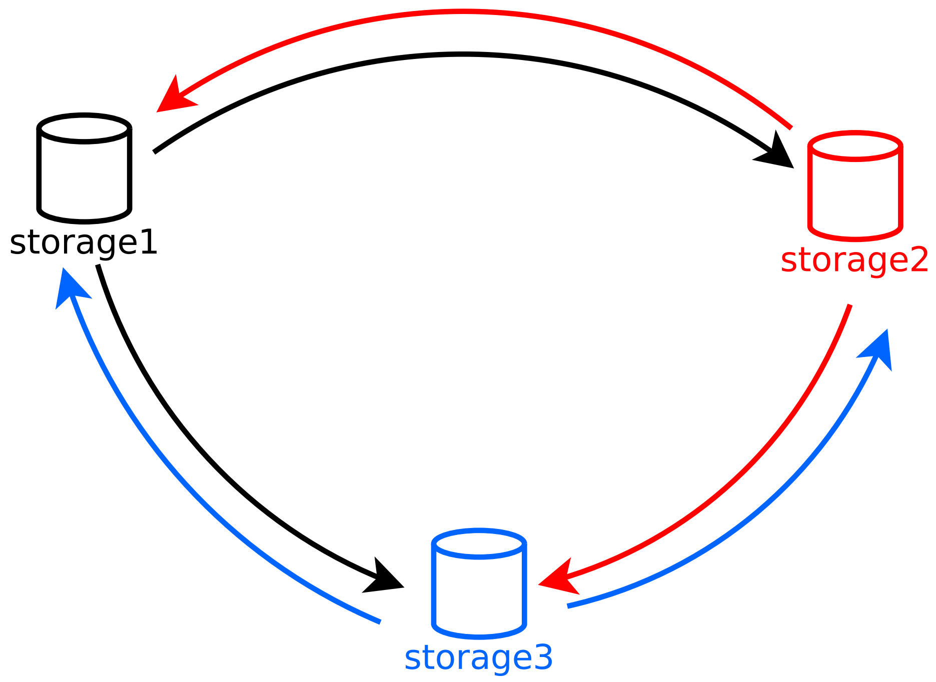
## systemd timer

```
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

```
storage1 ~ # 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

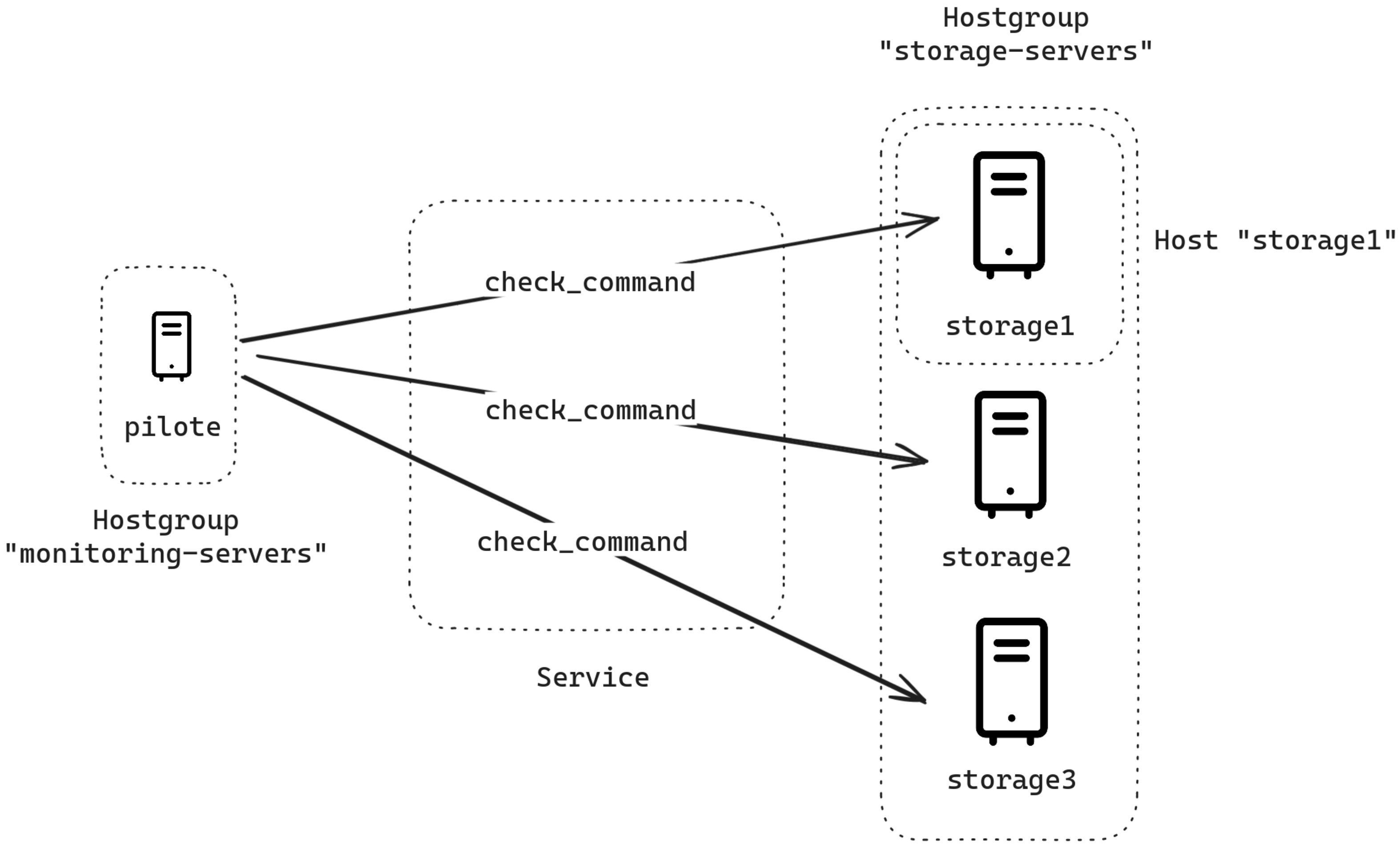
- Nagios Core
- Simple configuration files
- Web UI
- Plugins

# Welcome to `pilote` !

- Runs on Raspberry Pi
- Debian based distribution

# Components

- Hosts
- Hostgroups
- Services
- Notifications


Hostgroup
"storage-servers"
Host "storage1"
check_command
check_command
check_command
pilote
storage1
storage2
storage3
Hostgroup
"monitoring-servers"
Service

# 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

# Service configuration

```
define service {
    use                 home-service
    hostgroup_name      storage-servers
    service_description zfs_snapshots
    check_command       check_nrpe!check_zfs_snapshots
}
```

# NRPE agent

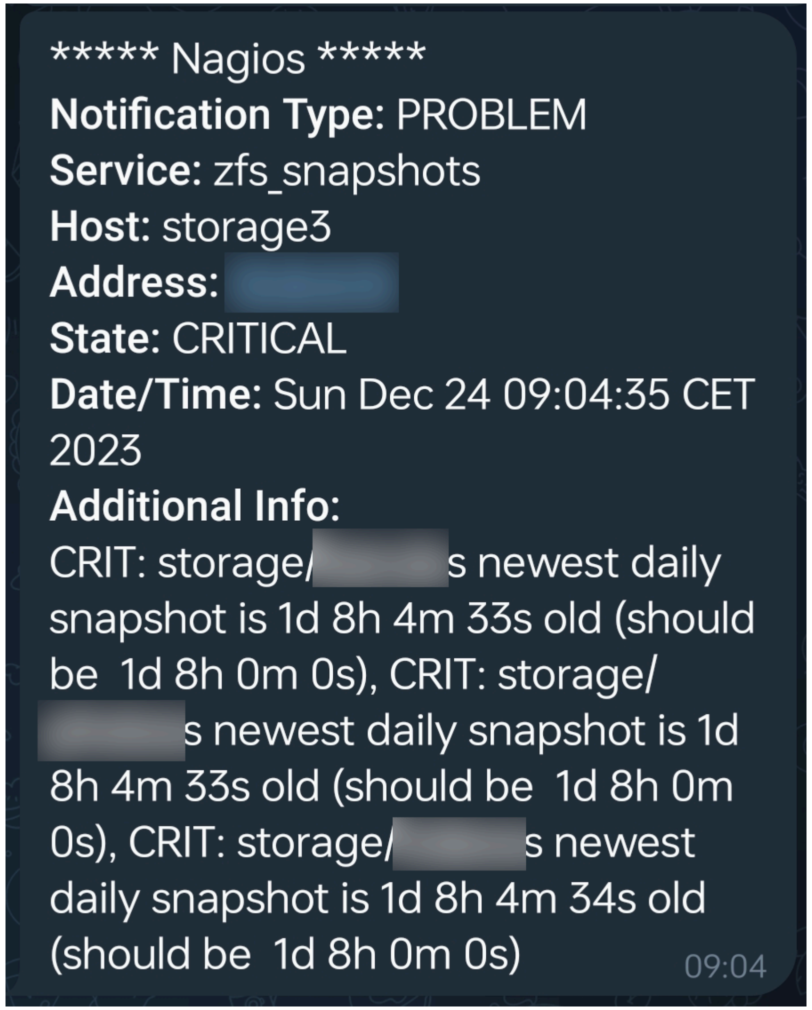
/etc/nagios/nrpe_local.cfg

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

# Notifications

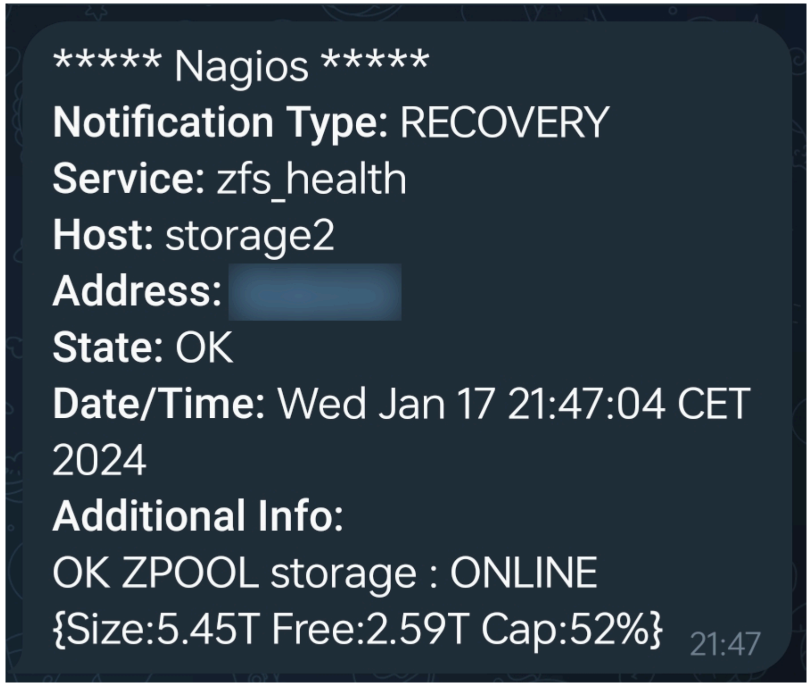
*Send Nagios notifications to a Telegram Messenger channel.*

notify-by-telegram



***** Nagios *****
**Notification Type:** PROBLEM
**Service:** zfs_snapshots
**Host:** storage3
**Address:** [redacted]
**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)

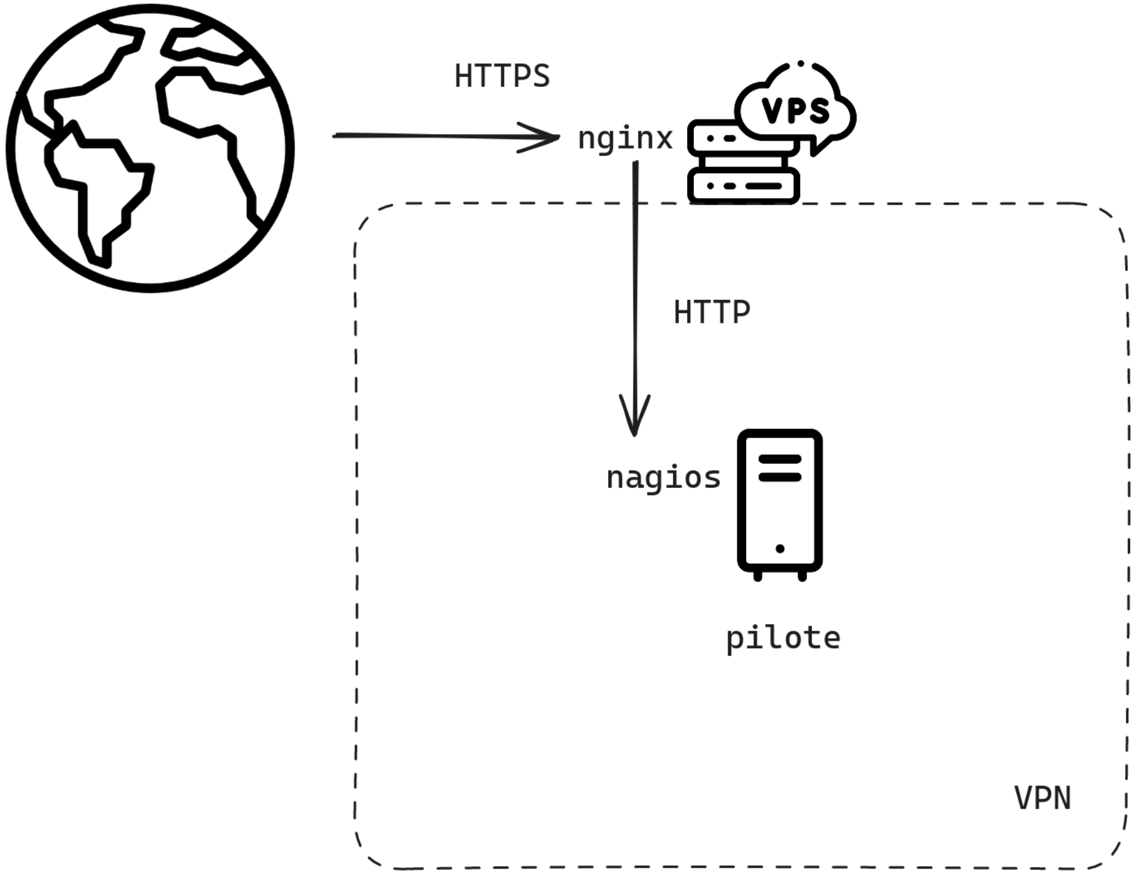
09:04

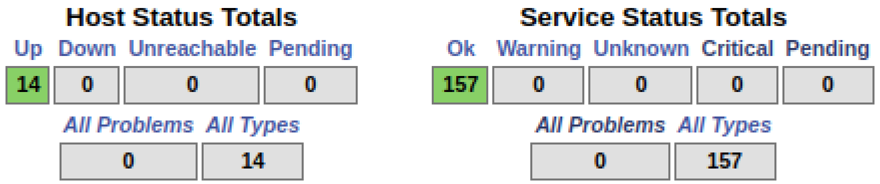


***** Nagios *****
**Notification Type:** RECOVERY
**Service:** zfs_health
**Host:** storage2
**Address:** [redacted]
**State:** OK
**Date/Time:** Wed Jan 17 21:47:04 CET 2024
**Additional Info:**
OK ZPOOL storage : ONLINE {Size:5.45T Free:2.59T Cap:52%} 21:47

# Web UI

# External access





**Host Status Totals**

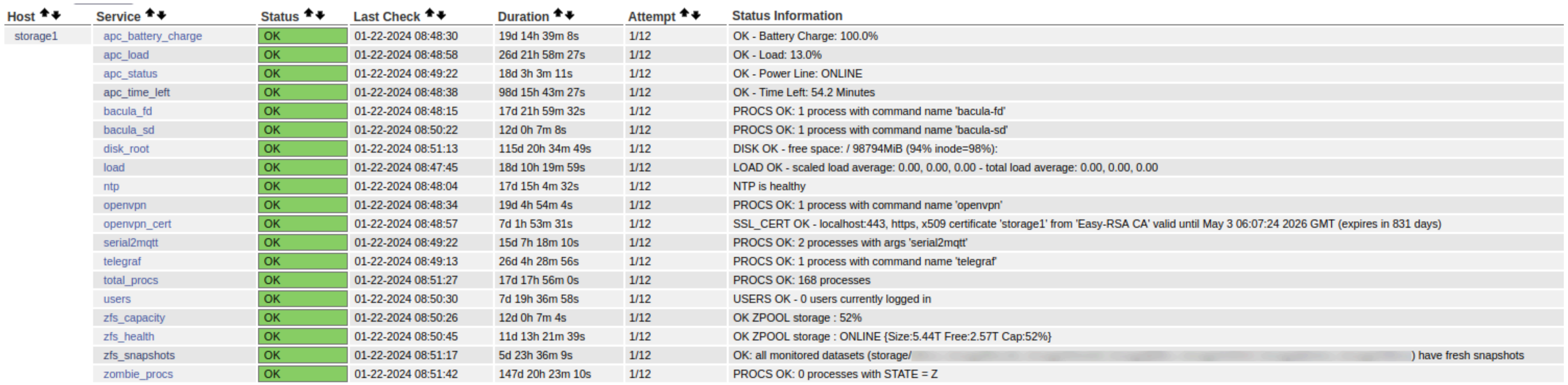
| Up | Down | Unreachable | Pending |
|---|---|---|---|
| 14 | 0 | 0 | 0 |

| *All Problems* | *All Types* |
|---|---|
| 0 | 14 |

**Service Status Totals**

| Ok | Warning | Unknown | Critical | Pending |
|---|---|---|---|---|
| 157 | 0 | 0 | 0 | 0 |

| *All Problems* | *All Types* |
|---|---|
| 0 | 157 |



| Host | Service | Status | Last Check | Duration | Attempt | Status Information |
|---|---|---|---|---|---|---|
| storage1 | apc_battery_charge | OK | 01-22-2024 08:48:30 | 19d 14h 39m 8s | 1/12 | OK - Battery Charge: 100.0% |
| | apc_load | OK | 01-22-2024 08:48:58 | 26d 21h 58m 27s | 1/12 | OK - Load: 13.0% |
| | apc_status | OK | 01-22-2024 08:49:22 | 18d 3h 3m 11s | 1/12 | OK - Power Line: ONLINE |
| | apc_time_left | OK | 01-22-2024 08:48:38 | 98d 15h 43m 27s | 1/12 | OK - Time Left: 54.2 Minutes |
| | bacula_fd | OK | 01-22-2024 08:48:15 | 17d 21h 59m 32s | 1/12 | PROCS OK: 1 process with command name 'bacula-fd' |
| | bacula_sd | OK | 01-22-2024 08:50:22 | 12d 0h 7m 8s | 1/12 | PROCS OK: 1 process with command name 'bacula-sd' |
| | disk_root | OK | 01-22-2024 08:51:13 | 115d 20h 34m 49s | 1/12 | DISK OK - free space: / 98794MiB (94% inode=98%): |
| | load | OK | 01-22-2024 08:47:45 | 18d 10h 19m 59s | 1/12 | LOAD OK - scaled load average: 0.00, 0.00, 0.00 - total load average: 0.00, 0.00, 0.00 |
| | ntp | OK | 01-22-2024 08:48:04 | 17d 15h 4m 32s | 1/12 | NTP is healthy |
| | openvpn | OK | 01-22-2024 08:48:34 | 19d 4h 54m 4s | 1/12 | PROCS OK: 1 process with command name 'openvpn' |
| | openvpn_cert | OK | 01-22-2024 08:48:57 | 7d 1h 53m 31s | 1/12 | SSL_CERT OK - localhost:443, https, x509 certificate 'storage1' from 'Easy-RSA CA' valid until May 3 06:07:24 2026 GMT (expires in 831 days) |
| | serial2mqtt | OK | 01-22-2024 08:49:22 | 15d 7h 18m 10s | 1/12 | PROCS OK: 2 processes with args 'serial2mqtt' |
| | telegraf | OK | 01-22-2024 08:49:13 | 26d 4h 28m 56s | 1/12 | PROCS OK: 1 process with command name 'telegraf' |
| | total_procs | OK | 01-22-2024 08:51:27 | 17d 17h 56m 0s | 1/12 | PROCS OK: 168 processes |
| | users | OK | 01-22-2024 08:50:30 | 7d 19h 36m 58s | 1/12 | USERS OK - 0 users currently logged in |
| | zfs_capacity | OK | 01-22-2024 08:50:26 | 12d 0h 7m 4s | 1/12 | OK ZPOOL storage : 52% |
| | zfs_health | OK | 01-22-2024 08:50:45 | 11d 13h 21m 39s | 1/12 | OK ZPOOL storage : ONLINE {Size:5.44T Free:2.57T Cap:52%} |
| | zfs_snapshots | OK | 01-22-2024 08:51:17 | 5d 23h 36m 9s | 1/12 | OK: all monitored datasets (storage/[redacted]) have fresh snapshots |
| | zombie_procs | OK | 01-22-2024 08:51:42 | 147d 20h 23m 10s | 1/12 | PROCS OK: 0 processes with STATE = 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"
```

influxdb™

*Scalable datastore for metrics, events and real-time analytics*

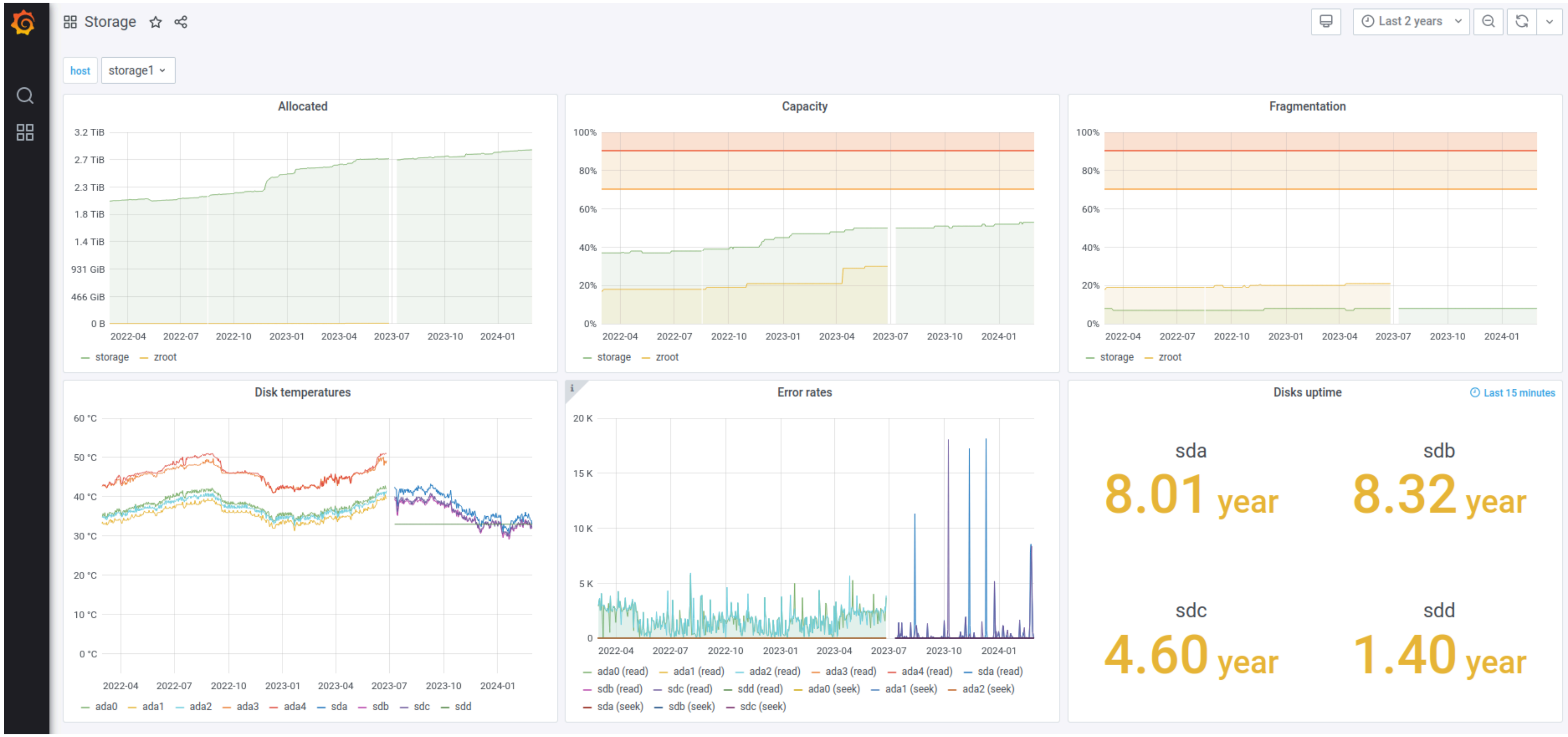
https://github.com/influxdata/influxdb

# Grafana

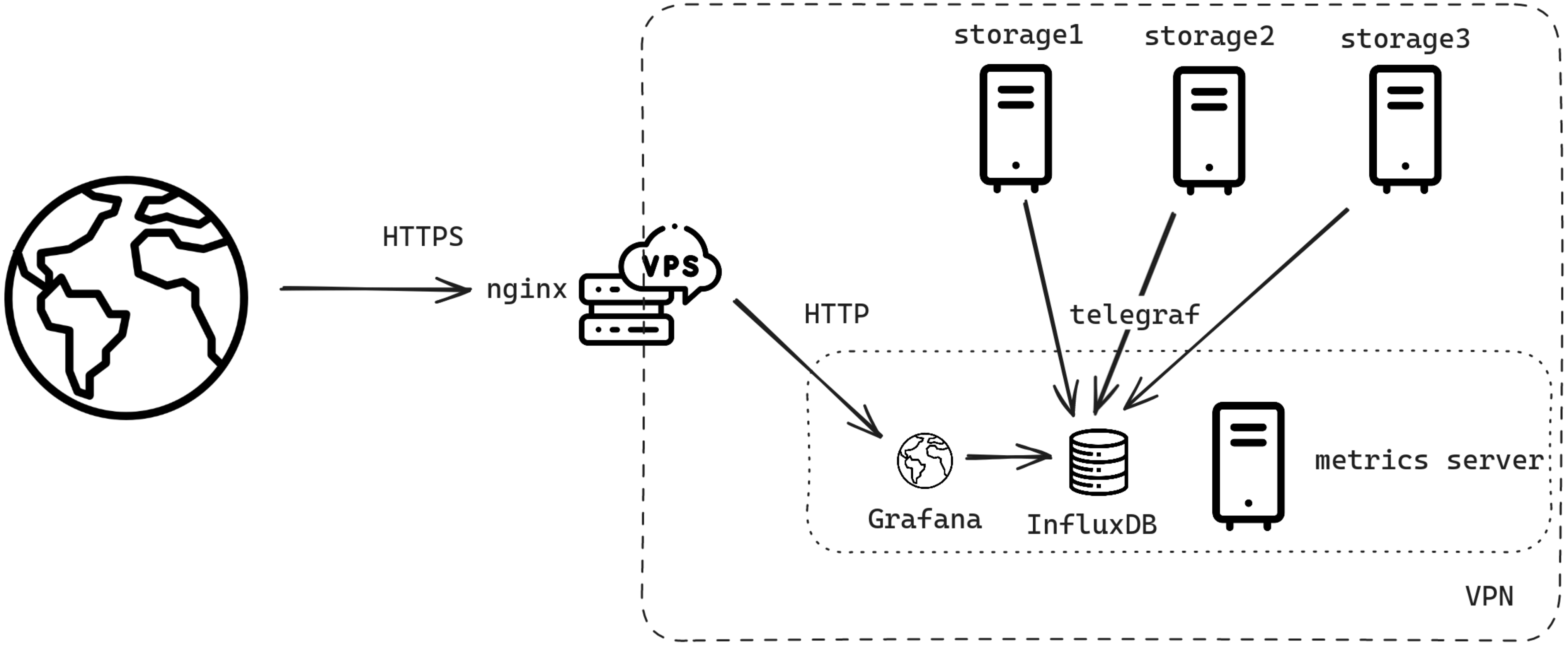
*The open-source platform for monitoring and observability*

https://github.com/grafana/grafana

# Grafana dashboard



# Overview


HTTPS
nginx
VPS
HTTP
storage1
storage2
storage3
telegraf
Grafana
InfluxDB
metrics server
VPN

# Notes

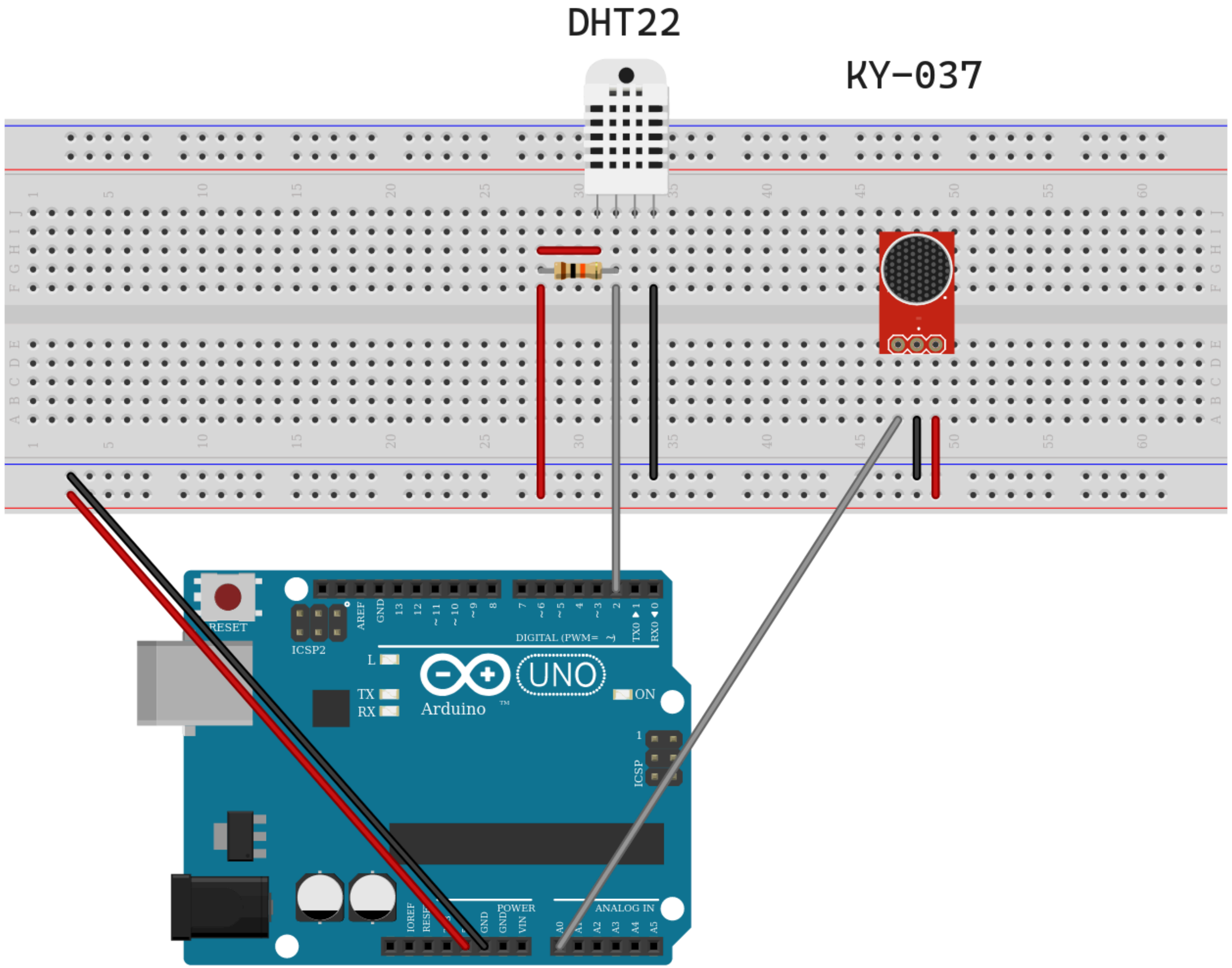
- Docker images available
  - InfluxDB, Grafana
- Not MicroSD friendly, prefer SSD or HDD
- Consider using Prometheus for the future

# Sensors

- Temperature
- Humidity
- Noise

# Hardware

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


DHT22
KY-037
RESET
ICSP2
AREF
GND
DIGITAL (PWM= ~)
TX
RX
L
ON
UNO
Arduino
ICSP
POWER
ANALOG IN

# 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

```
void setup()
{
    Serial.begin(9600);
    dht.begin();
}
```

# Main loop (1/2)

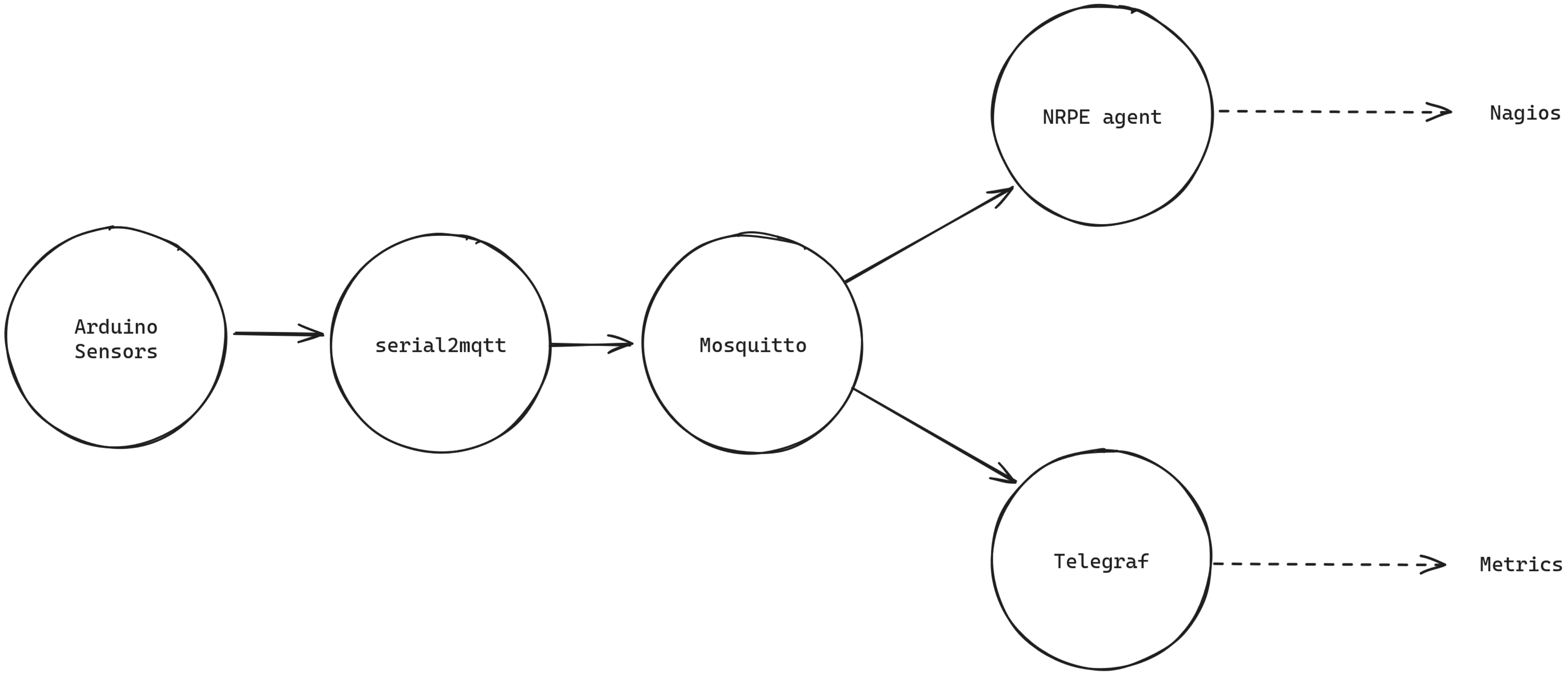
```
void loop()
{
    // sensors need some time to produce valid values
    delay(2000);

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

# Main loop (2/2)

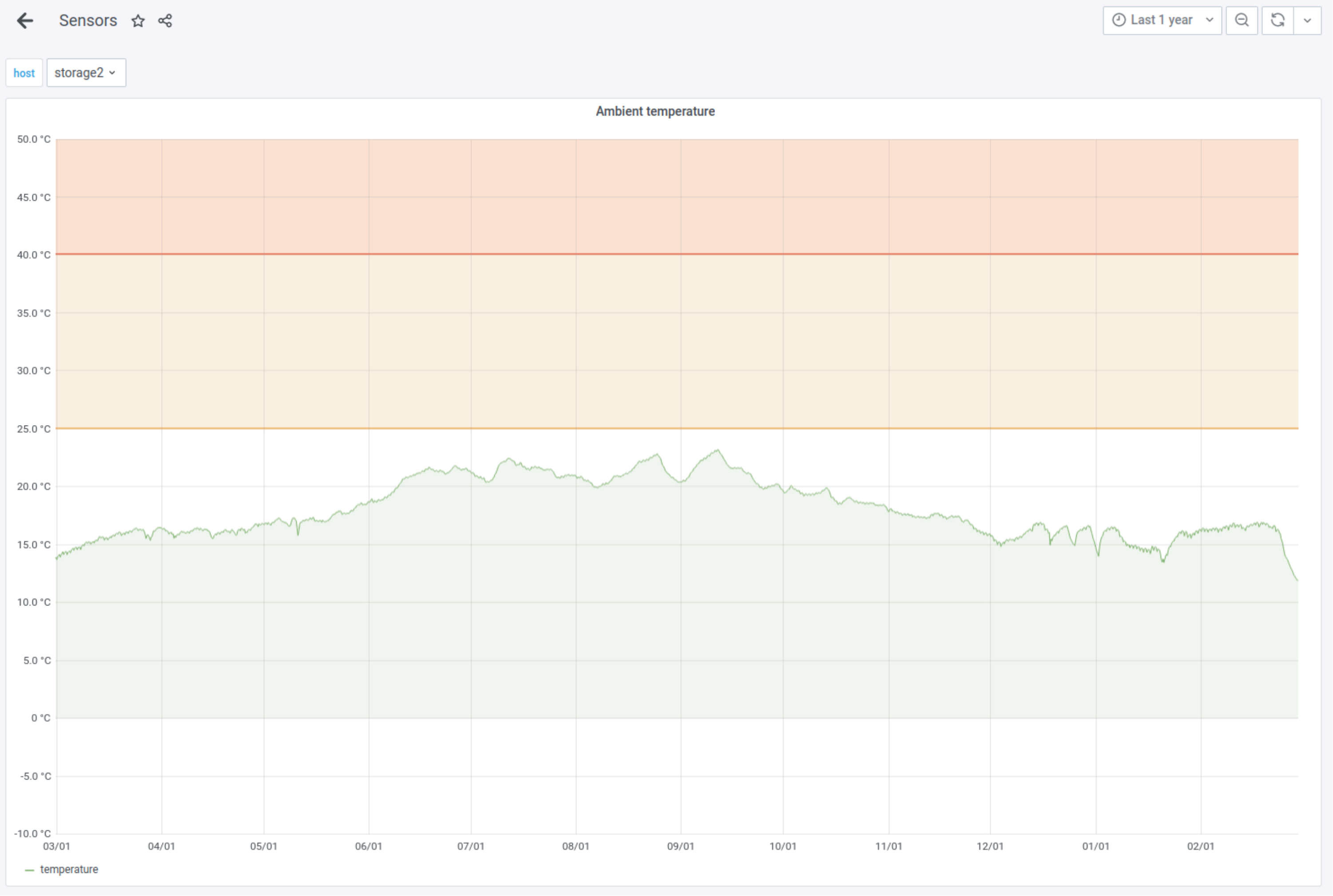
```
    // print "<humidity>,<temperature>,<sound>" (CSV-like)
    if (!isnan(h) && !isnan(t) && !isnan(s)) {
        Serial.print(h);
        Serial.print(",");
        Serial.print(t);
        Serial.print(",");
        Serial.println(s);
    }
}
```

# Multiplexing

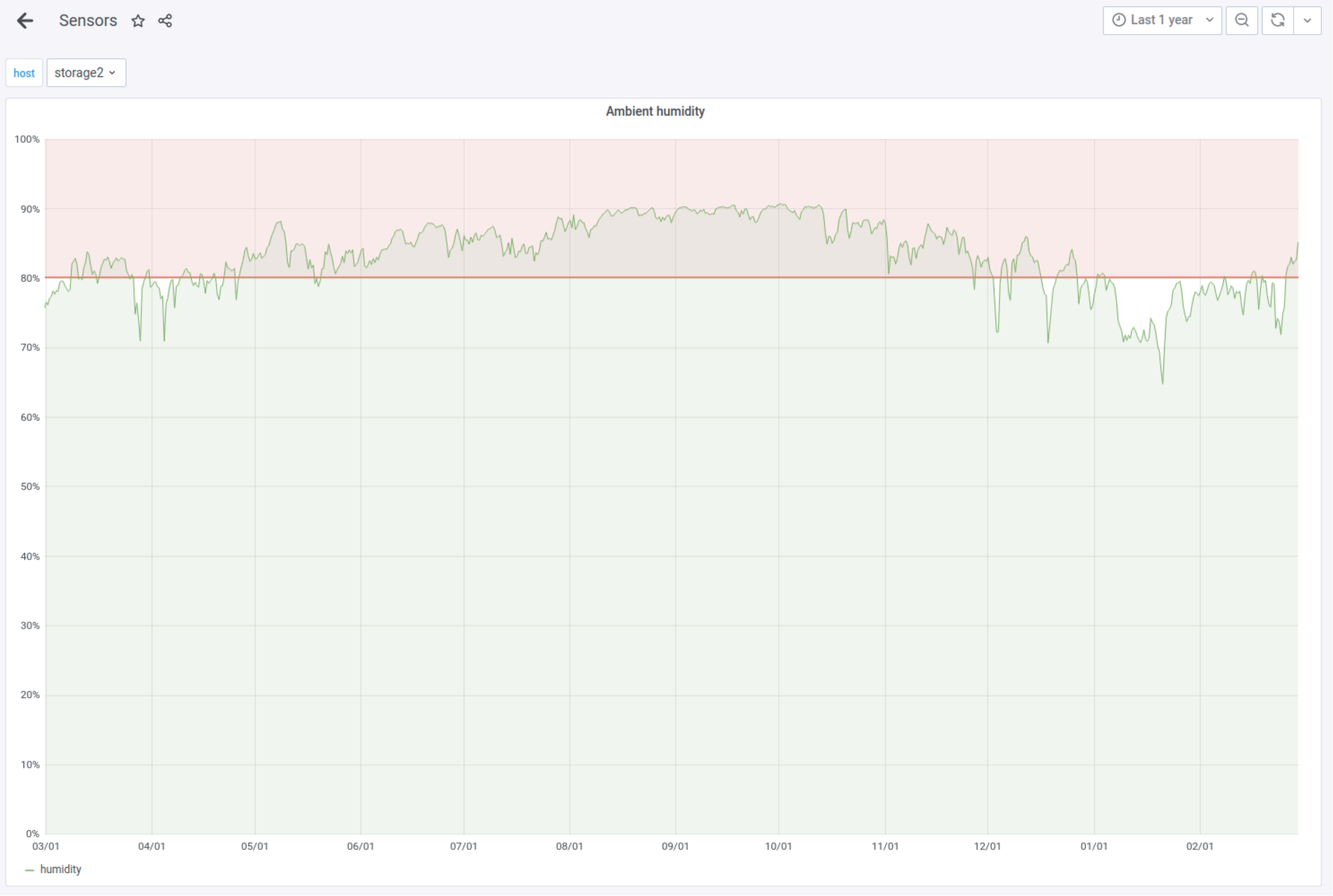
- Serial port can be accessed by only one program
- MQTT Broker (Mosquitto)
- serial2mqtt
- Nagios check-mqtt
- Telegraf mqtt_consumer


Arduino Sensors
serial2mqtt
Mosquitto
NRPE agent
Nagios
Telegraf
Metrics

# How is the temperature?

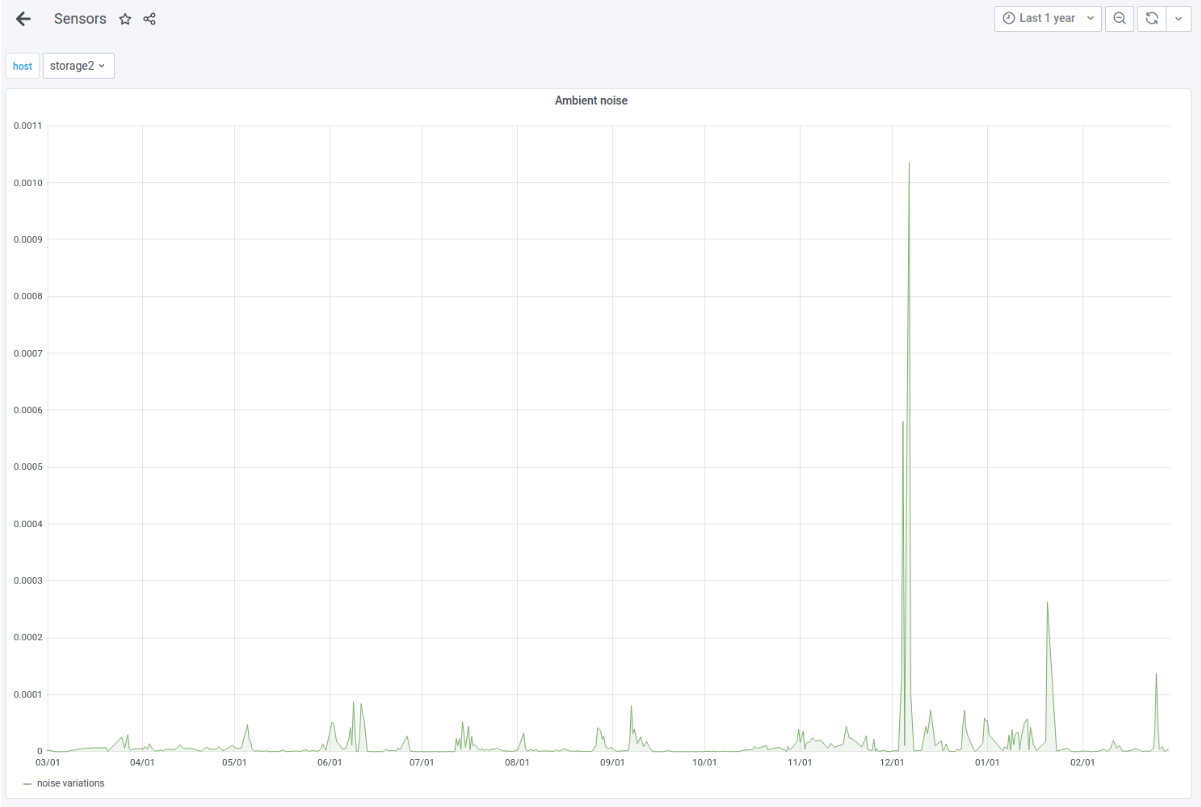


# Humidity

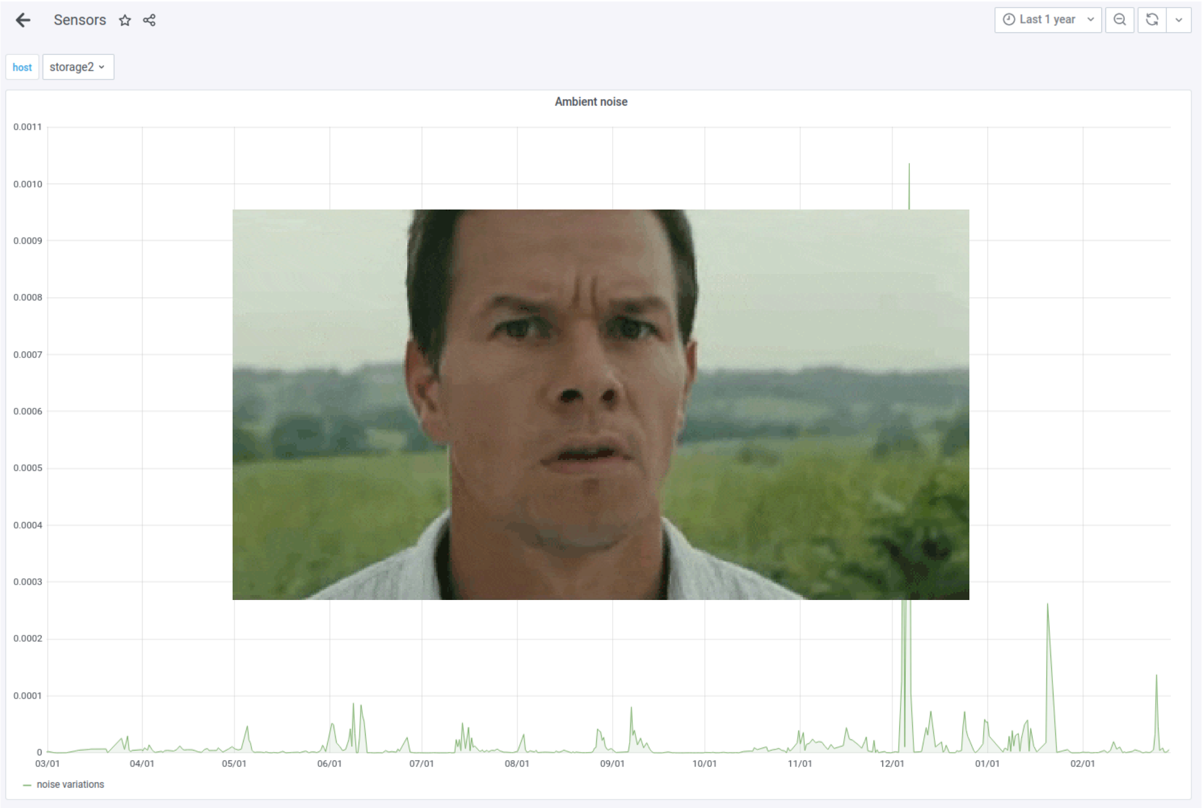

Sensors
Last 1 year
host
storage2
Ambient humidity
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
03/01
04/01
05/01
06/01
07/01
08/01
09/01
10/01
11/01
12/01
01/01
02/01
humidity

# Noise

# Noise


Sensors
Last 1 year
host
storage2
Ambient noise
0.0011
0.0010
0.0009
0.0008
0.0007
0.0006
0.0005
0.0004
0.0003
0.0002
0.0001
0
03/01
04/01
05/01
06/01
07/01
08/01
09/01
10/01
11/01
12/01
01/01
02/01
noise variations

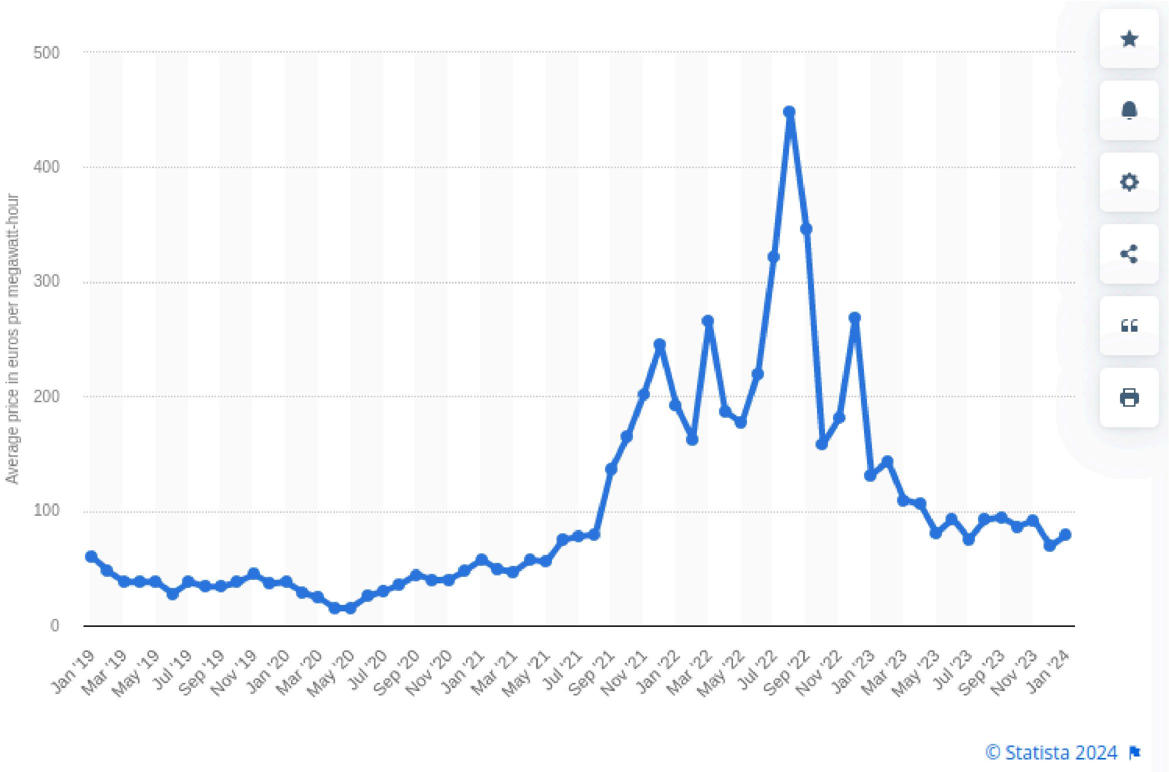
# Noise


Sensors
Last 1 year
host
storage2
Ambient noise
0.0011
0.0010
0.0009
0.0008
0.0007
0.0006
0.0005
0.0004
0.0003
0.0002
0.0001
0
03/01
04/01
05/01
06/01
07/01
08/01
09/01
10/01
11/01
12/01
01/01
02/01
noise variations

# Power consumption

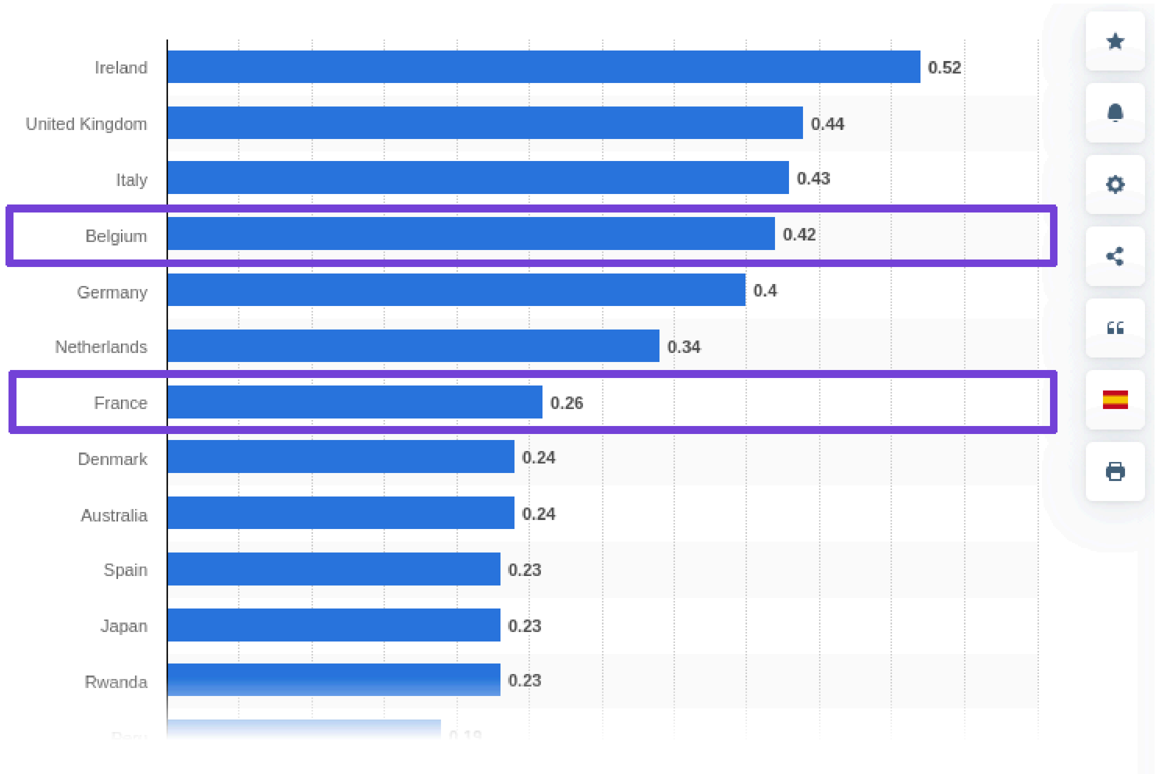
How much will it cost?

# Average monthly electricity wholesale price in Belgium from January 2019 to January 2024 (in euros per megawatt-hour)



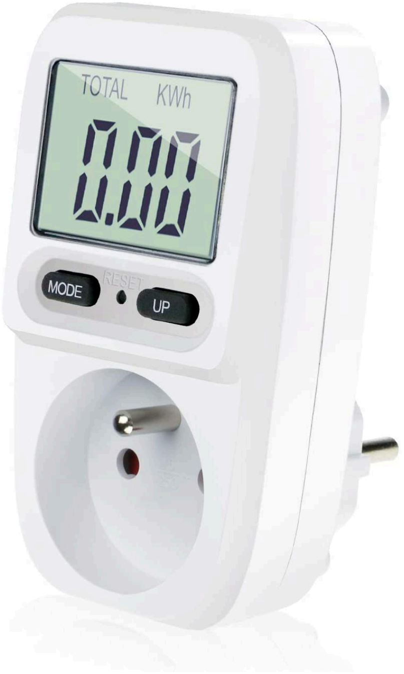
© Statista 2024

# Household electricity prices worldwide in June 2023, by select country (in U.S. dollars per kilowatt-hour)



© Statista 2024

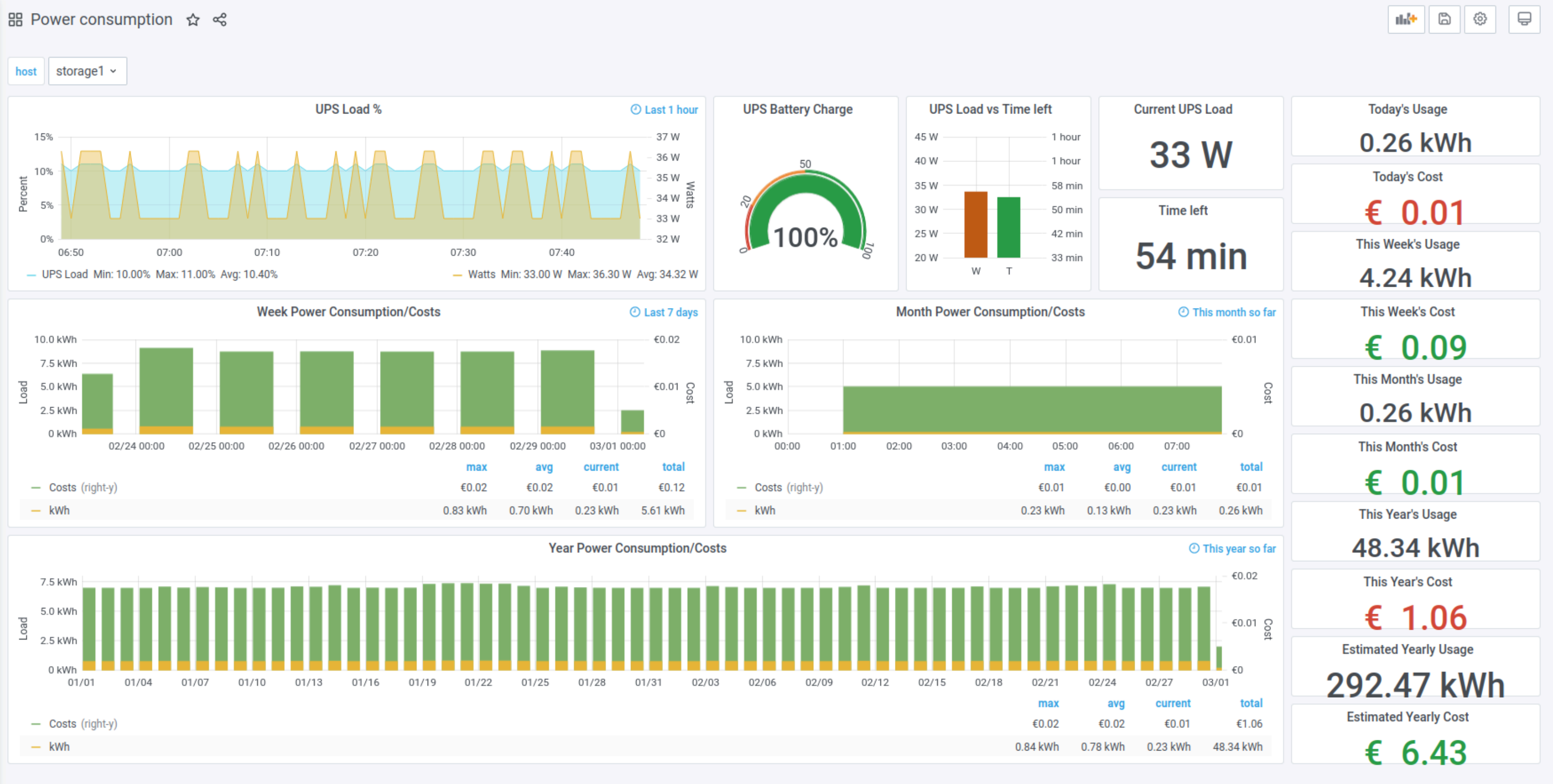
# Wattmeter



# 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



Power consumption

host storage1

UPS Load % — Last 1 hour

UPS Load Min: 10.00% Max: 11.00% Avg: 10.40%

Watts Min: 33.00 W Max: 36.30 W Avg: 34.32 W

UPS Battery Charge: 100%

UPS Load vs Time left

Current UPS Load: 33 W

Time left: 54 min

Today's Usage: 0.26 kWh

Today's Cost: € 0.01

This Week's Usage: 4.24 kWh

This Week's Cost: € 0.09

This Month's Usage: 0.26 kWh

This Month's Cost: € 0.01

This Year's Usage: 48.34 kWh

This Year's Cost: € 1.06

Estimated Yearly Usage: 292.47 kWh

Estimated Yearly Cost: € 6.43

Week Power Consumption/Costs — Last 7 days

| | max | avg | current | total |
|---|---|---|---|---|
| Costs (right-y) | €0.02 | €0.02 | €0.01 | €0.12 |
| kWh | 0.83 kWh | 0.70 kWh | 0.23 kWh | 5.61 kWh |

Month Power Consumption/Costs — This month so far

| | max | avg | current | total |
|---|---|---|---|---|
| Costs (right-y) | €0.01 | €0.00 | €0.01 | €0.01 |
| kWh | 0.23 kWh | 0.13 kWh | 0.23 kWh | 0.26 kWh |

Year Power Consumption/Costs — This year so far

| | max | avg | current | total |
|---|---|---|---|---|
| Costs (right-y) | €0.02 | €0.02 | €0.01 | €1.06 |
| kWh | 0.84 kWh | 0.78 kWh | 0.23 kWh | 48.34 kWh |

$7/y

# In real life

235
APC
Back-UPS Pro 550

APC
ZALMAN

236
APC
Back-UPS Pro 550
LAN1 LAN2 HDD ID FAIL

# Automation

# Failures happen

# Failures happen

MicroSD cards with I/O errors

# Failures happen

Flood or fire in the house

# Failures happen

# 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

```yaml
- 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

```yaml
- name: Deploy Syncoid script
  ansible.builtin.template:
    src: syncoid.sh.j2
    dest: /opt/syncoid.sh
    owner: zfs
    group: root
    mode: "0750"
```

# Template using Jinja2

```
#!/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 \
    {% if syncoid_source_bwlimit %}--source-bwlimit={{ syncoid_source_bwlimit }} {% endif %}
{% 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

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

# Upgrades

## upgrade.yml

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

## tasks/apt-upgrade.yml

```yaml
- 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