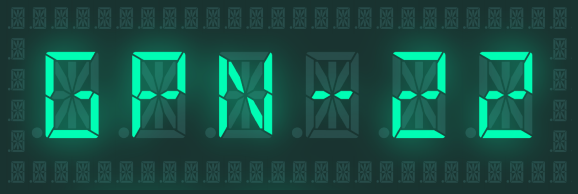
why the cloud is evil

An informed opinion and judgement of where we are heading with the cloud.

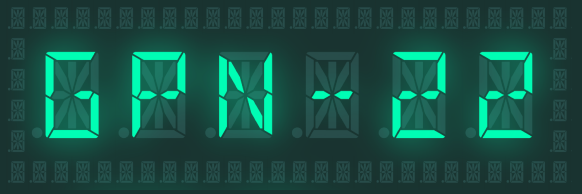
2024-05-30
https://fosstodon.org/@slink

/me



- Born when unix time fit into 28 bits
- Linux since ~1992 (kernel ~0.9.8 IIRC)
- MSc in Artificial Intelligence
- Set up two ISPs, ran a third
- Learned “everything” from FOSS
- Since 2009, Independent Developer and Consultant
- Varnish-Cache Maintainer
- Runs a small company

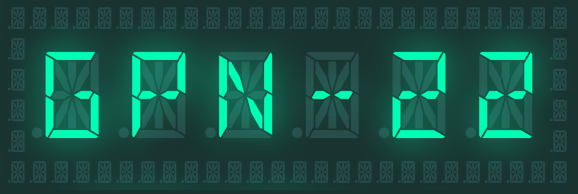
Disclaimer



- I am not a journalist
- I am bad at collecting references (evidence)

- This talk is partly based on *hypotheses*
- This talk is about my *informed opinion*
  - based on real world experience
- Do not expect any news
- Do your own research!
  - and prove me wrong, if I am

Context

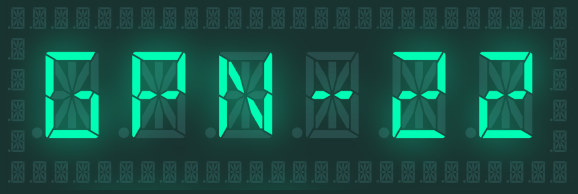


- I admire great technology
- *Cloud* is an old dream about resource sharing and fully automated provisioning
  - As a 28bit-epoch-person I still remember the Sun Grid Engine and related projects (N1, Ops Center, …)
- But ...

Scope

- What this talk is not about
- What it is about
- Technology vs. SAAS

This talk is NOT about



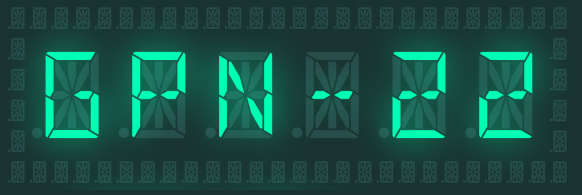
- ... people who
  - self-host kubernetes, docker swarm, openstack, ceph, minio, …
  - work at a hosting company providing webspace or applications,
  - run their own CDN on varnish-cache, nginx, traefik, ...
  - actively contribute to FOSS

This talk i

- People who

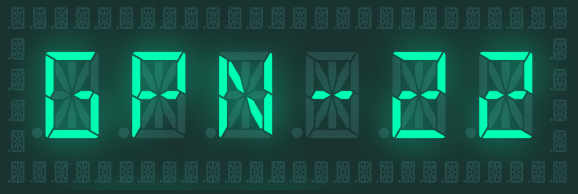
https://www.tiktok.com/@ventifriedchicken/video/7275421982366387502

This talk is ~~for~~ about



- **CAPSWAG**
  - **A**mazon **W**eb **S**ervices
  - **G**oogle **C**loud **P**latform
  - **A**zure

Devops

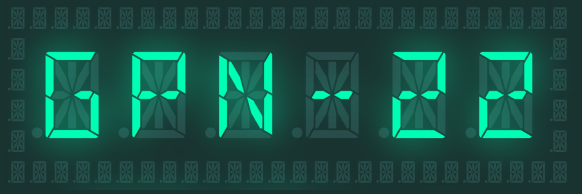


- Do you run your software
  - using *(semi) manual processes* (shell scripts, rsync, …) or
  - *fully automated* (ansible, puppet, salt, chef, helm, terraform...) ?
- Automation may be right for you, but beware of „one size fits all“ & dogmatism

Technology vs. SAAS

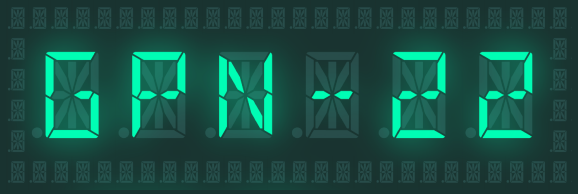
- It is *technology* (software) which brings the great features commonly associated with *cloud* (self-service, resource pooling, elasticity/scalebility, resilience)
- As long as good FOSS projects exist, you can just use the *technology* to your advantage
- You do not need a *service* to benefit from *technology*.
- Your *computer program* is the *same,* no matter where it runs

Cloud Native?



- A dogma by the Cloud Native Computing Foundation
  - scalable applications
  - in ... dynamic environments
  - (using, for example)
  - containers
  - service meshes,
  - microservices,
  - immutable infrastructure, and
  - declarative APIs

Cloud Native?



- Can your setup be Cloud Native outside the Cloud?
- Obviously yes!
- (you can also call “outside the cloud“ a “private cloud“, which helps the CNCF to claim to hold the universal truth)

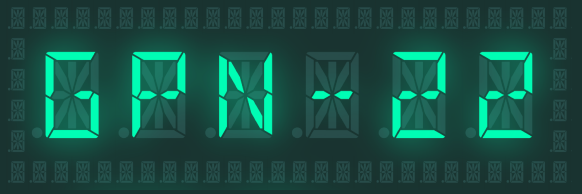
Cloud Myths

- “But it’s cheap“
  - I did a real world comparison in 2022

H/W

- kubernetes
- 7 worker nodes
  - 2 x Intel 8358 2,6 Ghz 32C / 64T, 11,2GT/s, 48MB Cache, DDR4-3200
  - 1.5 TB RAM
  - 2 x 7,68 TB SSD
  - 12 x 20TB HDD
- 5 mgmt nodes
  - Intel E-3334 3,4GHz 4C
  - 16GB RAM
  - 2 x 960GB SSD

vs. AWS



- 560 TiB S3
- 1164 GHz CPU
- 10214 GB memory
- EKS, S3 Data out, ELB ...

Numbers for 5yrs

- H/W hosted @ DC

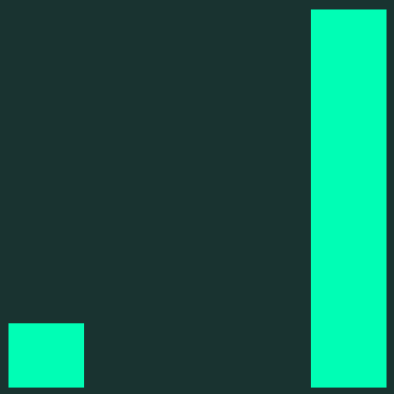
  ~8.500€ / month

- Break even at 17% resource usage
- Semi-fixed cost

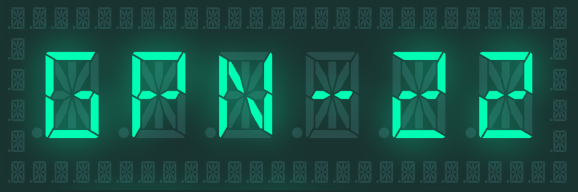
- AWS

  ~50.000€ / month

- Factor x5.8
- Dynamic cost

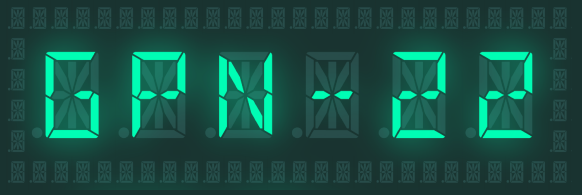


“But labor cost“



- Cloud marketing: “It’s all so easy that you can fire even the last competent IT person“
- Reality: Running hardware & infra is relatively easy, running the *application* is hard!
- My experience
  - Knowledge is still important
  - People just do *other* things
- (We will come back to this)

What others say



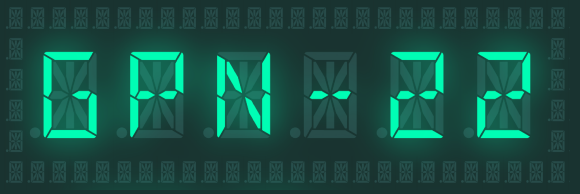
- David H Hansson (DHH) / hey.com:
  Save $7m over five years from our cloud exit
  - The Big Cloud Exit FAQ
- Ahrefs:

  How Ahrefs Saved US$400M in 3 Years by NO T Going to the Cloud

Design Issues

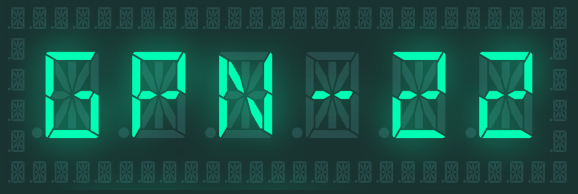
- Centralization
- Monoculture
- Anti-FOSS
- Vendor Lock-In
- Sustainability of knowledge
- Empowered engineers?
- Complexity

Centralization



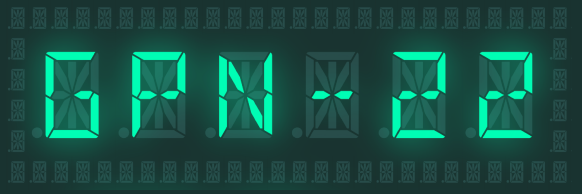
- Power of the internet is:
  - Decentralization
  - Cooperation & Competition
- What are the CAPSWAGs doing?
  - Massive Centralization
  - Kill all smaller companies
- “a perversion of the internet“
- Simply TOO BIG

Monoculture



- Gravitate towards “everything done the same way on one of three companies’ services“
- Dogmatism everywhere
  - Anything not *best practice* is considered wrong
  - Anything not *cloud native* is considered wrong
  - Anything not … is considered wrong
- If it fails, it fails BIG TIME
  - Google Cloud accidentally deletes $125 billion Australian pension fund

FOSS quick reminder

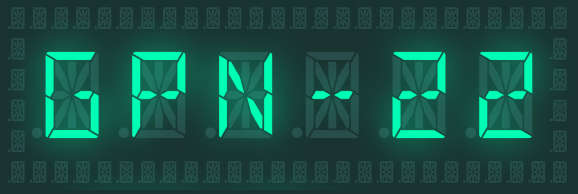


- FSF: The four essential freedoms
- **0: run the program as you wish**, for any purpose
- **1: study how the program works, and change it** so it does your computing as you wish
- **2: redistribute copies** so you can help others
- **3: distribute** copies of your **modified versions** to others

Anti-FOSS

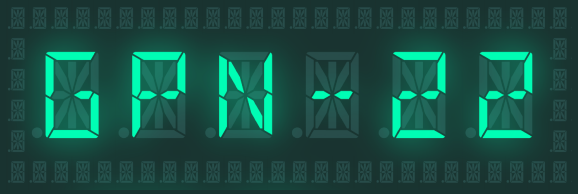
- “The Internet“ and Cloud providers would not exist without Free Open Source Software
  - Virtually everything of relevance is built on Linux, BSD, GNU and thousands of FOSS projects
- CAPSWAGs turn FOSS into SAAS
  - Still the same software behind the service, but you lose all the freedoms
  - To me, cloud is “Windows 95“

Anti-FOSS 2



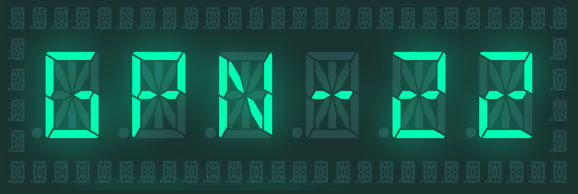
- Most of the Software CAPSWAGs build *on top of* FOSS is not itself FOSS
- Marketing: Many of the cool performance achievements CAPSWAGs brag about are actually FOSS achievements which they provision
- “Standards“: The “good“ Internet Protocols were created collaboratively and with consensus
  - But CAPSWAGs “innovate first“ and then let the rest of the world swallow whatever hack they came up with (S3, HTTP/2)

Vendor Lock in



- CAPSWAGs promote high abstraction
  - “serverless“
  - “cloudless“
- High abstraction = difficult migration
- → Windows 95
- Vendor Lock-in looks like not a big deal – until it does:
  Cloudflare took down our website after trying to force us to pay 120k$ within 24h

What do you learn today?

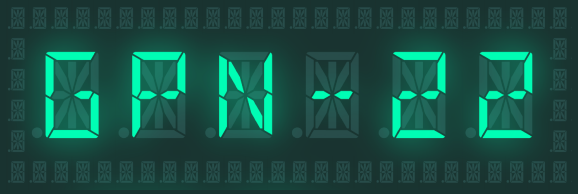


- FOSS knowledge will be beneficial forever
  - You learn how software and computers work
- CAPSWAG knowledge is wasted on proprietary system
  - You learn how to use “fill out their order form”
  - → Windows 95

How do you work?

- With FOSS, you learn, analyze and fix
  - In principle, you can fix everything yourself (if you understand enough)
- With CAPSWAG, you learn what the vendor wants you to
  - If anything goes wrong, you are back to „open a service case“
  - → Windows 95

Empowered? Really?



- CAPSWAGs tell you that they empower engineers
  - Yes, they may if compared to the old it-department style
  - But that is DevOPS, not cloud!
- CAPSWAGs tell you that you do not need to understand the complicated detail (“low code“)
  - So now your PM can run the site?
  - Does it make you happy to be kept stupid?

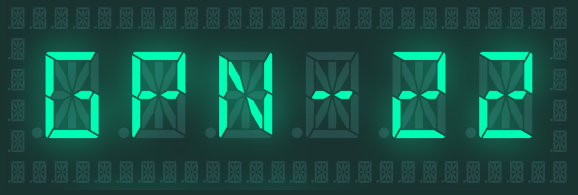
The dumb engineers

- So we leave all the complicated stuff to CAPSWAGs
- How do we control them? Where are the checks and balances?
  - Their systems are closed, we can not properly review them

Hidden Complexity

- Cloud Services are not simple
- They might look simple
- But they are actually more complex
- And all / most of the complexity is hidden from you

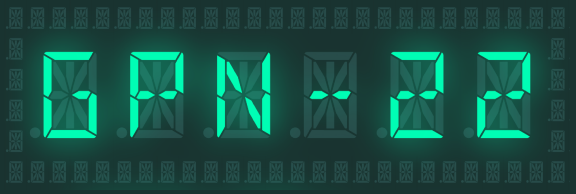
To the managers



- If you only hire “cloud engineers“, who do you get?
- If you invest in “cloud trainings“, how good an investment is it?
- Good engineers are going to leave
- Organizations are going to lose know-how

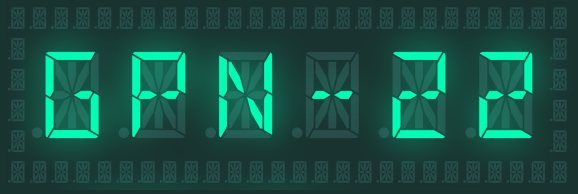
- Also: Replace Organizations with States

$€¥



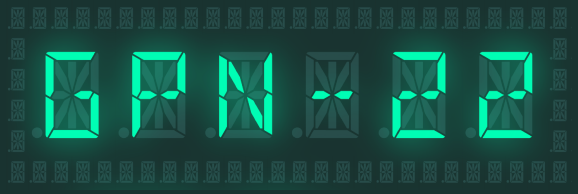
- CAPSWAGs turn huge profits
- They need to massively over-provision to keep their promise of “endless resources“
- And you think you are getting a good deal?
- The business model:
  - Buy and run computers
  - Build more and more abstract layers
  - Sell the computer power for x-fold the cost
  - Tell people they are too stupid to do this themselves

But you save on admins



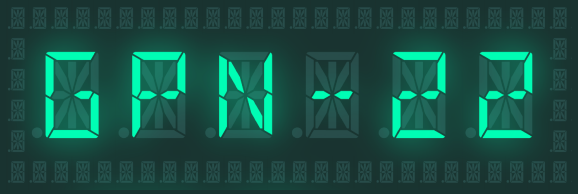
- Yes? Really?
- For simple deployments, *maybe*
- But CAPSWAG need a lot of custom (= non-sustainable) knowledge
- So now instead of paying people to run systems, you are paying people to make them fit the CAPSWAG

Flexibility or Chaos?



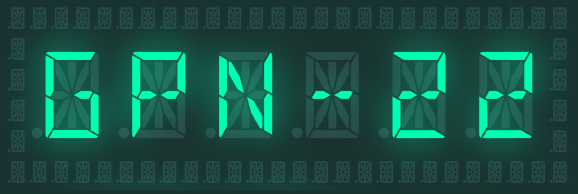
- Devs manage everything themselves and choose whatever they like
  - 1000s of “serverless“ deployments talking to each other
  - Data leaks
- High fluctuation in many teams
  - Deployment configs age quickly
  - Who maintains them?
- More Cloud to the rescue!
  - Just use another tool to manage the “security“
    - REALLY?

Which problem do they solve?



- CAPSWAGs tell you “Oh running computers is sooo complicated, there is backup and security and ooooh it’s way too hard“
  - BUT IT IS NOT
  - Just hire some smart people & let them do their work
  - They will solve the other hard problems too!

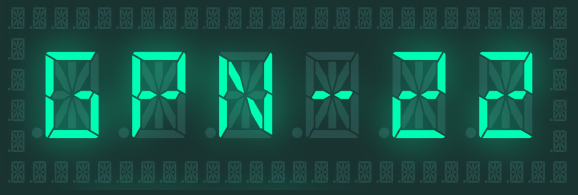
Running Computers



- Today’s servers are **VERY** reliable
  - Basically swap a disk once in a while
  - And leave the rest to the vendor service
- Today’s software is **VERY** fault tolerant
  - Much of the innvoation “the cloud“ builds upon is actually in the FOSS you can run yourself

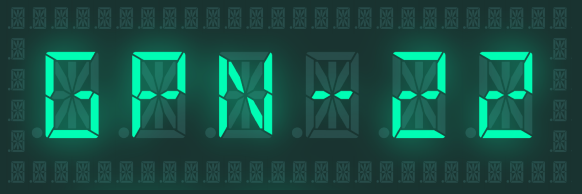
Practical aspects

So, when/how to „cloud“



- Use it when the main benefits apply, when
  - you need <u>massive</u> scalability like >1000x
    - You probably don’t
  - you run prototypes with lots of resources for a short time only
  - your traffic / resource requirements vary A LOT over time
- For backup / redundancy
- Never use just one provider
  - Always “multicloud“

Last resort



- If your primary system is “cloud“, **REPLICATE DATA ON-SITE**
  - **You should be able to hold your data in your hands**

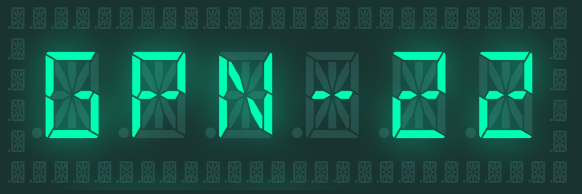
When not to „cloud“

- Your base load is probably cheaper to run with a friendly local hosting company
- For most of the sites I know, the base load is quite constant
  - Relatively low user base and/or
  - Web Caching

Where are we heading?

- Unless we convince (more) decision makers to **change course** ...

I think we will...

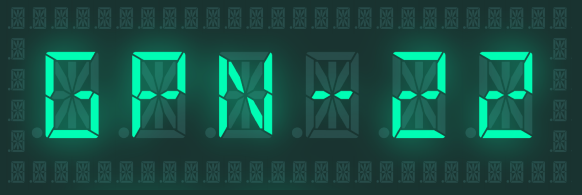


- become totally dependent on a small number of companies from the cart^Woligopoly
- see massive outages on a global scale
- see “too big to fail” companies fail
  - and be bailed out by society
- see the knowledge inequality rise
- as engineers, become YAML monkeys

End with a positive note

- If we manage to not **deliver ourselves to the cloud** ...

I think we have the chance to



- spend our limited time with interesting work
- on FOSS for the benefit of all
- stay knowledgeable and competent
- and keep our digital sovereignity on all levels

Thank you!