Infrastructure Internship

May-Jun 2024 @ i-bitz company limited

Pongpeera Wongprasitthiporn's presentation
Computer Engineering, KMITL.

Table of contents

An overview of things I learned at i-bitz!

01

Python Scripting

Coding, but less painful

The problem

Create a Python program to set up Docker Engine and initialize a MongoDB replica set while...

- Getting system memory size
- Allow setting number of containers
- Getting % of system memory allowed

Python results overview

A single-file Python script that...

- Users can set parameters on the top of the file
- Uses **subprocess** to install dependencies and the Docker engine
- Gets amount of system memory via **psutil**
- Runs specified amount of MongoDB containers
- Uses **MongoClient** from **pymongo** to initialize a replica set

Mongo Replica Initializer (Python)

on my self-hosted Forgejo instance

02

Go programming

A lot of improvements

The problem

Create a Python program to set up Docker Engine and initialize a MongoDB replica set while...

- Getting system memory size
- Allow setting number of containers
- Getting % of system memory allowed

But this time in Go.

Rewrite It In ~~Rust~~ Go

- Go is a new language, at least for me.
- The Python prototype was already there. I just need a Go equivalent.
- I couldn’t get the Go Docker Client to work on the first days.
- I’m going to support more server-oriented Linux distributions. This means more testing is required.
- Need to use **?directConnection=true** which wasn’t ideal for production.

Anyways, let’s go straight to the results.

Mongo Replica Initializer (Go)
on my self-hosted Forgejo instance

Simple Compose Runner

A simple CLI tool to run docker-compose at a specified location, without having to change directories again and again.

- Run a compose in current directory **./main**
- Run a compose in current directory, verbose output **./main -v**
- Run at specified location **./main /path/to/repo**
- Run at specified location, verbose output **./main -v /path/to/repo**

Simple Compose Runner

on my self-hosted Forgejo instance

03

Kubernetes

From K3s to Helm charts

Why Kubernetes?

Kubernetes is a tool to help manage containerized workflows/apps (aka Pods), distribute them between multiple nodes for high availability, load-balance them and manage internal networks – so Pods can communicate between each other, even if they’re on different machines!

So many Kubernetes out there…

- I started with **K3s**, installed Kubernetes Dashboard but the web console didn’t want to connect to the cluster.
- I moved to **K8s**, this time the Dashboard works, but the control plane dies when the host’s LAN IP gets changed (e.g. switching between different Wi-Fi networks).
- I finally moved to **minikube** on my dev laptop. *More on this later.*

My experience with minikube (1/6)

Installation was simple, I just have to install a single RPM file (as a Fedora user) and make sure Docker is installed – since I'll be using the Docker driver that will run the cluster in a Docker container.

When the cluster is ready, I installed the Dashboard and Ingress just by running **minikube addons enable dashboard** and **minikube addons enable ingress**

Both of them were installed successfully, time to deploy some apps!

My experience with minikube (2/6)

There are 2 ways to deploy apps *that I've tried* – Applying "deployment" YAML files manually and using the Helm package manager.

I'll be using Helm to install **Invidious**, a 3rd party ad-free YouTube frontend.

First, set up the repos.

```
$ helm repo add invidious https://charts-helm.invidious.io
$ helm repo update
```

And simply deploy it with

```
$ helm install invidious invidious/invidious
```

My experience with minikube (3/6)

Make sure the invidious service exists. It should be already set up by Helm.

```
$ kubectl get svc
NAME                       TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)    AGE
invidious                  ClusterIP   10.98.234.39   <none>        3000/TCP   2d5h
invidious-postgresql       ClusterIP   10.103.35.98   <none>        5432/TCP   2d5h
invidious-postgresql-hl    ClusterIP   None           <none>        5432/TCP   2d5h
kubernetes                 ClusterIP   10.96.0.1      <none>        443/TCP    3d5h
```

Notice the port 3000/tcp on the service “invidious”. We’ll need this later.

My experience with minikube (4/6)

Now let’s expose it to the host via Ingress! Create an ingress.yml containing these:

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: invidious-ingress
spec:
  rules:
    - http:
        paths:
          - pathType: Prefix
            path: /
            backend:
              service:
                name: invidious
                port:
                  number: 3000
```

My experience with minikube (5/6)

Make sure the information in correct, then apply the Ingress.

```
$ kubectl apply -f ingress.yaml
```

Give it some time to start. On my system it took about 30 seconds.

Now get the list of Ingresses on the system.

```
$ kubectl get ingress
NAME                CLASS   HOSTS   ADDRESS         PORTS   AGE
invidious-ingress   nginx   *       192.168.49.2    80      2d5h
```

Open http://192.168.49.2 in your favorite browser.

INVIDIOUS

search

LOG IN

POPULAR TRENDING

Default
Music
Gaming
Movies

Here - Official Trailer (HD)

Sony Pictures Entertainment

Shared 20 hours ago **1M views**

The Acolyte - re:View

RedLetterMedia

Shared 1 day ago **830K views**

VINO TINTO (Video Oficial) - Peso Pluma, Natanael Cano, Gabito Ballesteros

Peso Pluma

Shared 10 hours ago **572K views**

LISA - ROCKSTAR (MV Teaser)

LLOUD Official

Shared 1 day ago **5.4M views**

asdfm

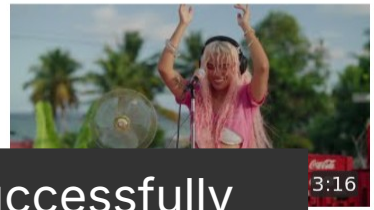
TomSk

Shared 1 day ago **2M views**

Shared 17 hours ago **320K views**

Rick Beato

Shared 1 day ago **1M views**



KAROL G

Shared 1 day ago **2.4M views**

Invidious on local Kubernetes cluster is now successfully installed. Enjoy your videos! (6/6)

04

Honorable mentions

Supporting side projects

Honorable mentions

During my time at i-bitz company limited, there are many other interesting things I’ve learned. Let’s go through the best ones *briefly*.

Website load testing/benchmarking

```
[sasha@alpine ~]$ wrk -t 8 -c 200 -d 10s --latency https://www.techtransthai.org
Running 10s test @ https://www.techtransthai.org
  8 threads and 200 connections
  Thread Stats   Avg      Stdev     Max   +/- Stdev
    Latency    80.26ms   54.73ms   1.05s    92.07%
    Req/Sec   231.23    119.77   430.00     54.50%
  Latency Distribution
     50%   64.02ms
     75%   79.71ms
     90%  124.50ms
     99%  337.07ms
  17451 requests in 10.02s, 222.24MB read
Requests/sec:   1741.87
Transfer/sec:     22.18MB
[sasha@alpine ~]$ wrk -t 8 -c 200 -d 10s --latency https://c4c.techtransthai.org
Running 10s test @ https://c4c.techtransthai.org
  8 threads and 200 connections
  Thread Stats   Avg      Stdev     Max   +/- Stdev
    Latency   204.09ms  140.08ms   1.38s    88.21%
    Req/Sec    83.96     45.88   242.00     67.05%
  Latency Distribution
     50%  162.38ms
     75%  270.57ms
     90%  357.31ms
     99%  737.40ms
  6592 requests in 10.02s, 233.93MB read
  Socket errors: connect 0, read 0, write 0, timeout 6
Requests/sec:    657.98
Transfer/sec:     23.35MB
```

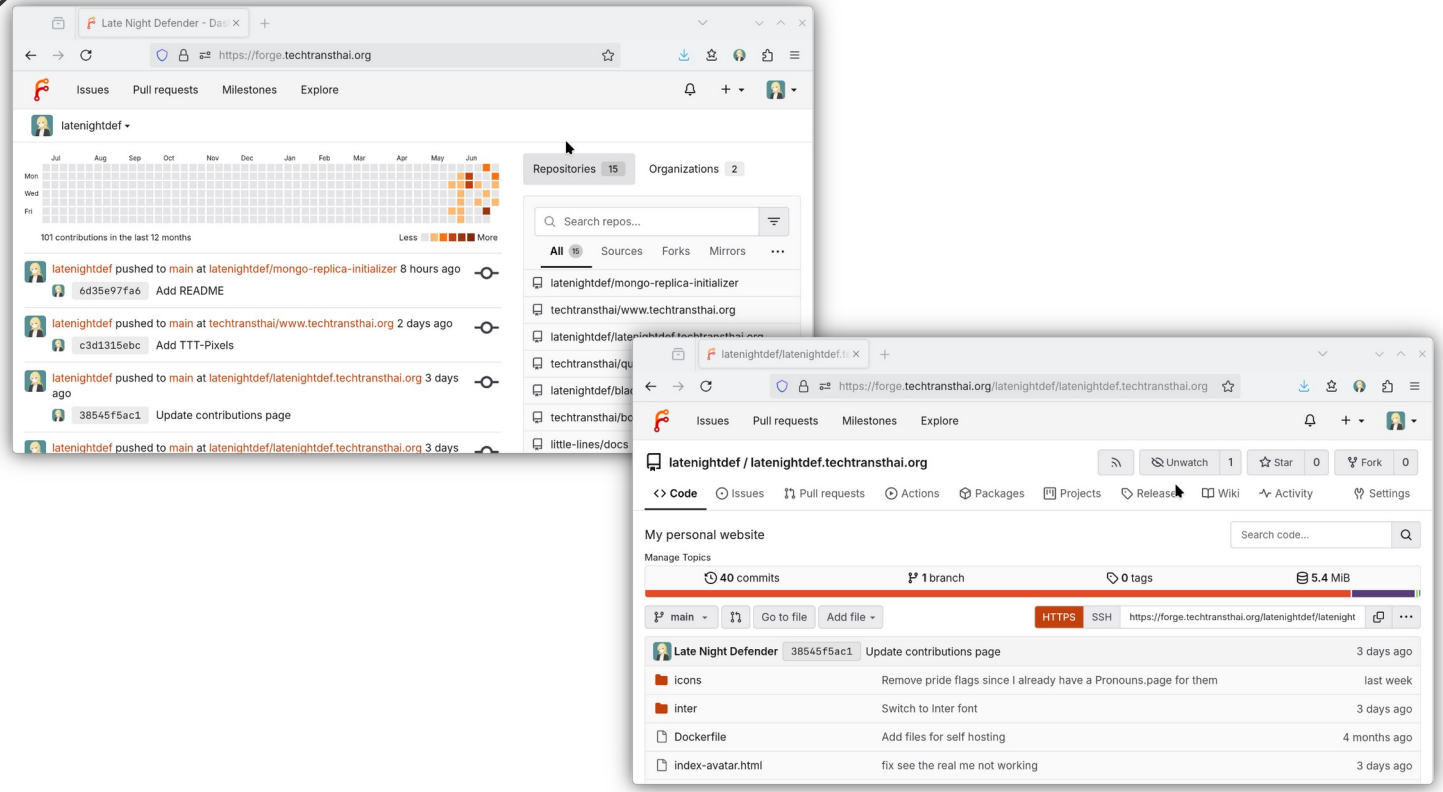
Running **wrk** at 8 threads, 200 connections for 10 seconds against my 2 websites. The top one runs on AWS while the bottom one runs on a home/residential fiber network.

Self-hosted Git Forge

I run a self-hosted Forgejo instance, so I can have complete control over all of my repos and easily migrate the whole forge if anything goes south.

Forgejo is a fork of Gitea that went live when the Gitea project was taken by the for-profit Gitea Ltd.

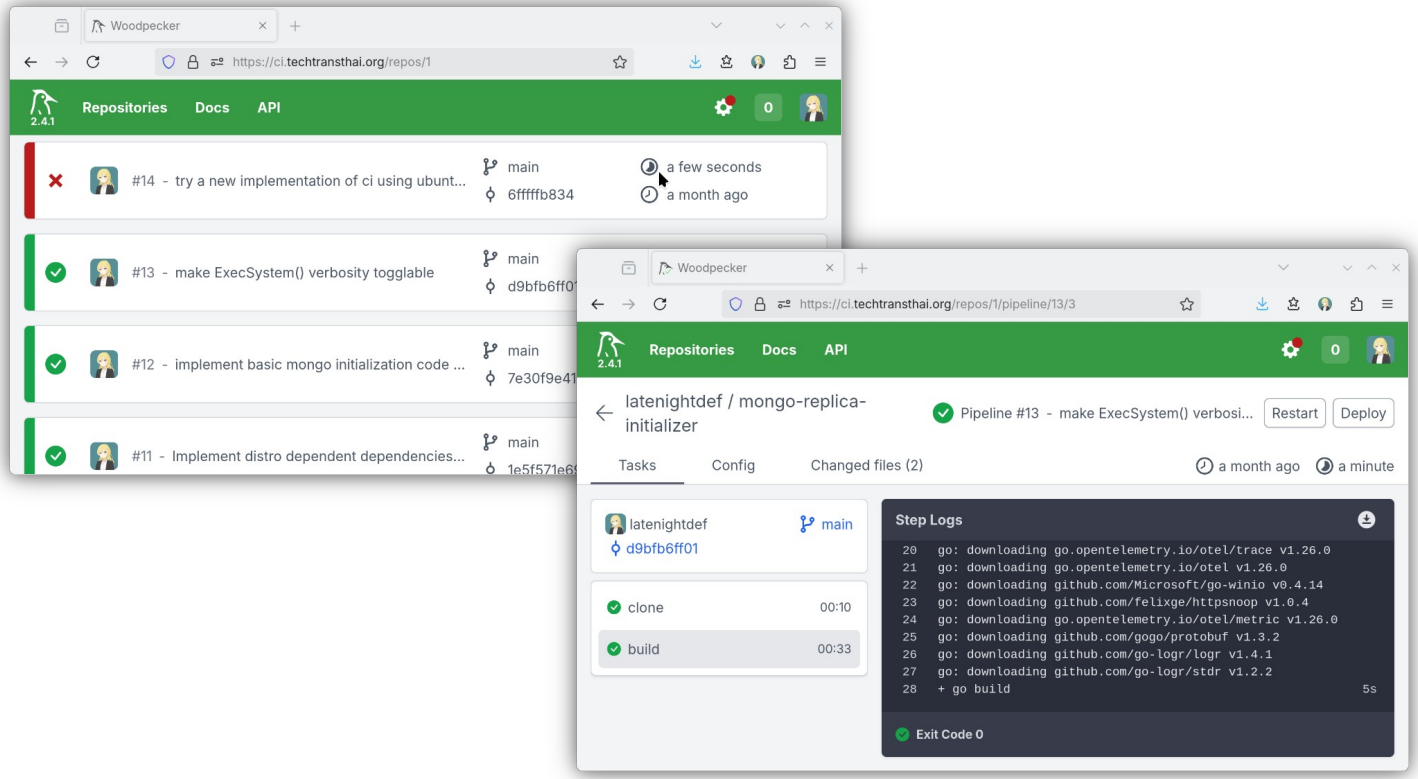
You can see my Forgejo instance in action at TechTransThai Forge


Late Night Defender - Das
https://forge.techtransthai.org
Issues
Pull requests
Milestones
Explore
latenightdef
Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr May Jun
Mon
Wed
Fri
101 contributions in the last 12 months
Less More
latenightdef pushed to main at latenightdef/mongo-replica-initializer 8 hours ago
6d35e97fa6 Add README
latenightdef pushed to main at techtransthai/www.techtransthai.org 2 days ago
c3d1315ebc Add TTT-Pixels
latenightdef pushed to main at latenightdef/latenightdef.techtransthai.org 3 days ago
38545f5ac1 Update contributions page
latenightdef pushed to main at latenightdef/latenightdef.techtransthai.org 3 days
Repositories 15
Organizations 2
Search repos...
All 15
Sources
Forks
Mirrors
latenightdef/mongo-replica-initializer
techtransthai/www.techtransthai.org
techtransthai/qu
latenightdef/bla
techtransthai/bc
little-lines/docs
latenightdef/latenightdef.tu
https://forge.techtransthai.org/latenightdef/latenightdef.techtransthai.org
Issues
Pull requests
Milestones
Explore
latenightdef / latenightdef.techtransthai.org
Unwatch 1
Star 0
Fork 0
Code
Issues
Pull requests
Actions
Packages
Projects
Releases
Wiki
Activity
Settings
My personal website
Manage Topics
Search code...
40 commits
1 branch
0 tags
5.4 MiB
main
Go to file
Add file
HTTPS
SSH
https://forge.techtransthai.org/latenightdef/latenight
Late Night Defender 38545f5ac1 Update contributions page 3 days ago
icons Remove pride flags since I already have a Pronouns.page for them last week
inter Switch to Inter font 3 days ago
Dockerfile Add files for self hosting 4 months ago
index-avatar.html fix see the real me not working 3 days ago

Self-hosted CI

I run a self-hosted Woodpecker CI, which is a fork of last free Drone CI before the relicense from Apache 2.0 to a proprietary license. I've attempted to use this with the Mongo Replica Initializer, but failed since the Mongo Replica Initializer did not work in a DinD'ed environment.

You can check out my CI attempt for Mongo Replica Initializer here


Woodpecker
https://ci.techtransthai.org/repos/1
2.4.1
Repositories
Docs
API
0
#14 - try a new implementation of ci using ubunt...
main
6fffffb834
a few seconds
a month ago
#13 - make ExecSystem() verbosity togglable
main
#12 - implement basic mongo initialization code ...
main
7e30f9e41
#11 - Implement distro dependent dependencies...
main
Woodpecker
https://ci.techtransthai.org/repos/1/pipeline/13/3
2.4.1
Repositories
Docs
API
0
latenightdef / mongo-replica-initializer
Pipeline #13 - make ExecSystem() verbosi...
Restart
Deploy
Tasks
Config
Changed files (2)
a month ago
a minute
latenightdef
main
d9bfb6ff01
clone
00:10
build
00:33
Step Logs
20 go: downloading go.opentelemetry.io/otel/trace v1.26.0
21 go: downloading go.opentelemetry.io/otel v1.26.0
22 go: downloading github.com/Microsoft/go-winio v0.4.14
23 go: downloading github.com/felixge/httpsnoop v1.0.4
24 go: downloading go.opentelemetry.io/otel/metric v1.26.0
25 go: downloading github.com/gogo/protobuf v1.3.2
26 go: downloading github.com/go-logr/logr v1.4.1
27 go: downloading github.com/go-logr/stdr v1.2.2
28 + go build
5s
Exit Code 0

Transactional email services (1/2)

I need a transactional email service for my Git Forge, especially when it comes to Git-related notifications so I can check them out in my main inbox.

This required an SMTP server.

I started out by trying to self-host a Postal instance on my EC2 server. But I found out later that EC2 IPs are banned and blacklisted almost in all spam filtering services, due to an unfortunate thing that spammers like to buy instances and use it for spamming.

Transactional email services (2/2)

Time to look for transactional email services.

I started with Amazon SES, but stuck in a sandbox and services are limited.

I checked out Postmark, but I need a company email to register on it.

I checked out MailerSend, but it only allow 100 emails as a trial.

I ended up using Amazon SES, since it’s just me and a few people I can verify emails with anyway.

I went ahead to my Cloudflare DNS dashboard and added DKIM/DMARC stuff to my domain. Now I have a working transactional email system.

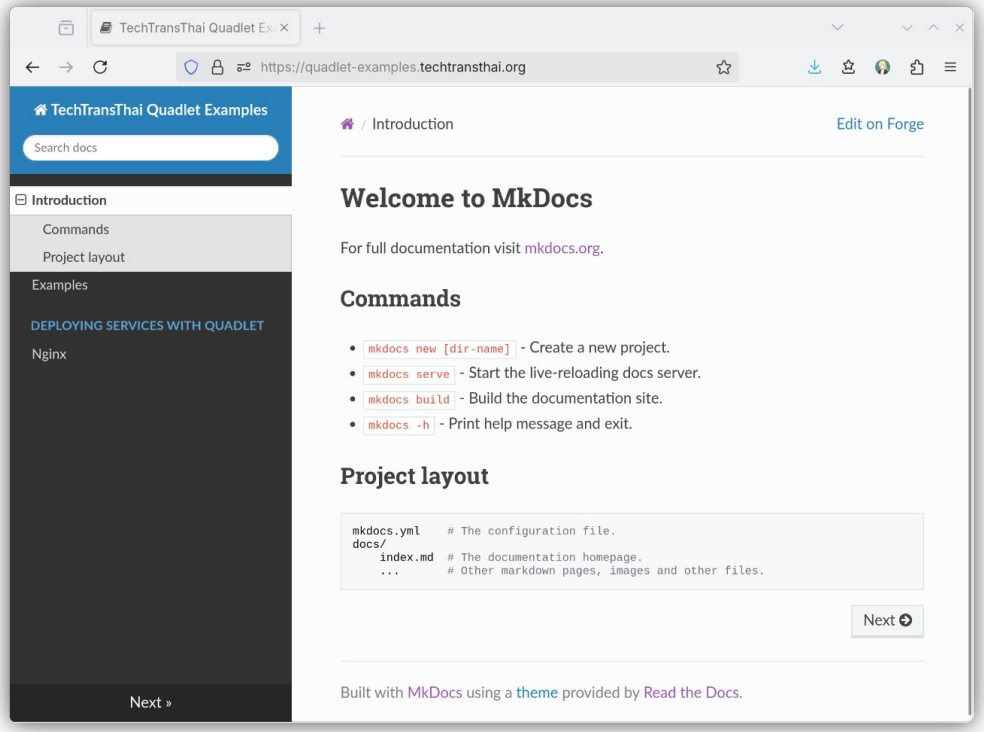
Docker Hub alternatives

To not be tied into Docker’s ecosystem, I tried 2 other services.

Red Hat Quay.io – The workflow is quite different, since the server running the registry is the build server by default, listening to webhooks from Git repositories when pushed onto. You can also self-host Quay yourself too!

Forgejo container registry – This is where the power of self-hosted Forgejo comes in! I can just docker login and push my images in a familiar way.

Documentation framework



My MkDocs site, hosted in a custom-bulit nginx Docker container.
This will be used for future projects' docs.

A container-optimized OS for K8s (1/2)

I use Linux exclusively these days, from my little Raspberry Pi to headless PCs running as home servers. Most of them are from the Red Hat family.

As I explored Kubernetes, something has caught my attention.

*“Fedora CoreOS is an automatically updating, minimal, monolithic, container-focused operating system, designed for clusters but also operable standalone, **optimized for Kubernetes but also great without it.** Its goal is to provide the best container host to run containerized workloads securely and at scale.”*

A container-optimized OS for K8s (2/2)

Container-focused systems aren’t anything new to me. In fact, I’ve been using them on and off since about Fedora Silverblue 33-34 days. We’re now at Fedora 40.

So I went ahead and booted up a new Fedora CoreOS virtual machine to see how different it is to traditional server/cloud-oriented distributions.

The K3s installer downloads stuff from the internet and tried to use normal Fedora’s YUM to install dependencies. It failed since YUM isn’t available for these atomic systems.

However, upstream K8s was available in the official sources and can be installed easily.

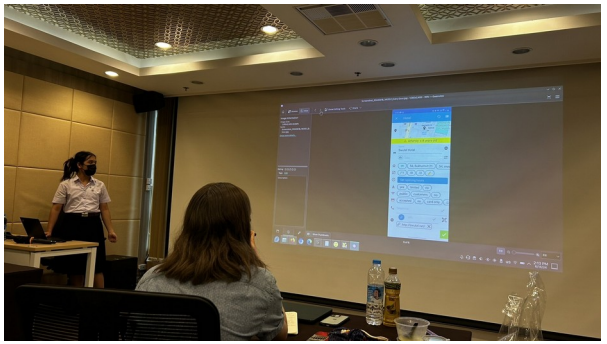
BONUS: Special events (1/2)

- On May 27, 2024, I was interviewed by the company’s marketing team. It was about how I use social media platforms, search engines, deciding factors for a software and trivial stuff.
- On June 5, 2024, I participated in the Vallaris training. This became an inspiration for my computer engineering graduate project at KMITL.

BONUS: Special events (2/2)

- On June 18, 2024, I participated in the Youth Mapper event, along with other interns and i-bitz staff at SWU. I had a presentation session about my 3-year experience contributing to the OpenStreetMap project

- I talked about 5 of the tools I used, from web apps to mobile ones.
- I talked about how can the OSM data be used, such as navigation, a11y and more.



Trivia

All serious things aside, I also learned **new modes of transportation** in Bangkok! Including BMTA, private EV buses and the old good diesel passenger trains!

Learning these new modes of transportation helped me save up to...

182 Baht/day

By replacing Airport Rail Link (plus parking fees) with SRT Eastern Line,
and replacing BTS Skytrain with classic BMTA buses.
From 210 Baht/day down to 28 Baht/day

Thanks

Do you have any questions?

I care about free software, decentralization, open data and open standards.
Made with LibreOffice on Linux BTW!