

Freifunk Tag 2017

# Freifunk in Containern

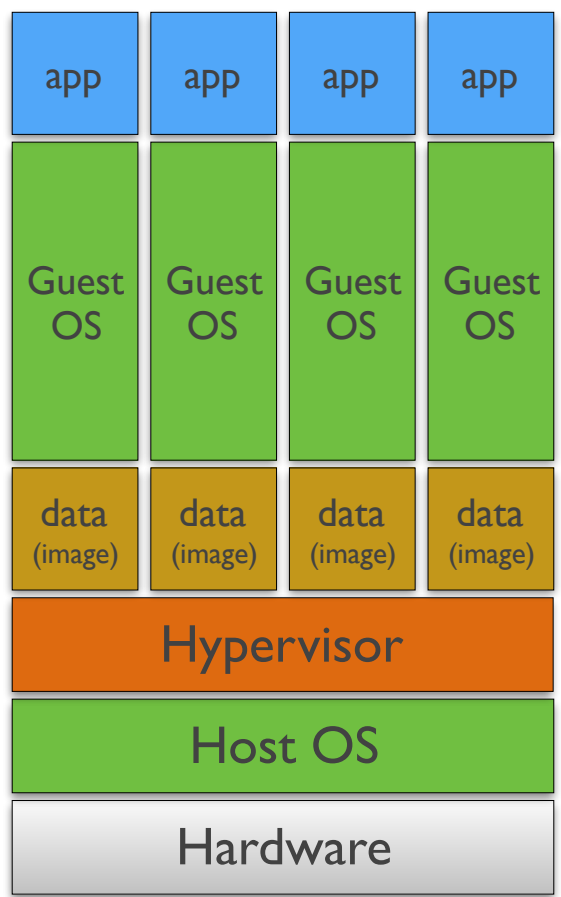
nico@freifunk-hennef.de



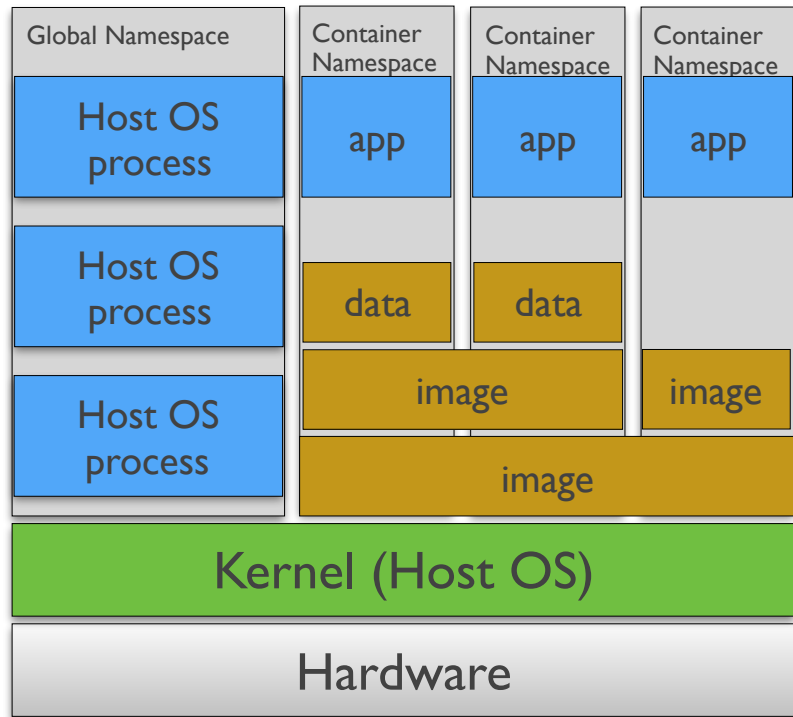
# Container?

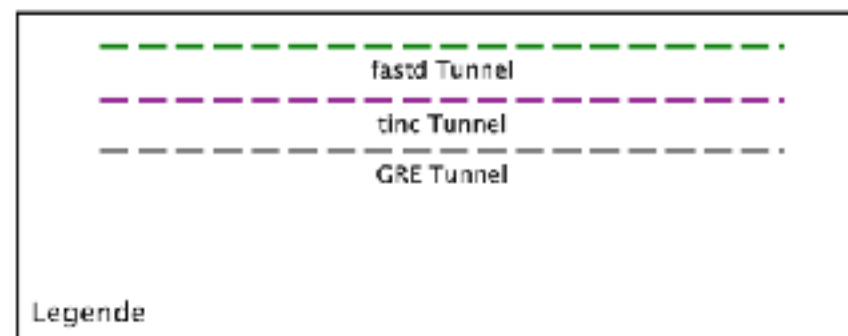
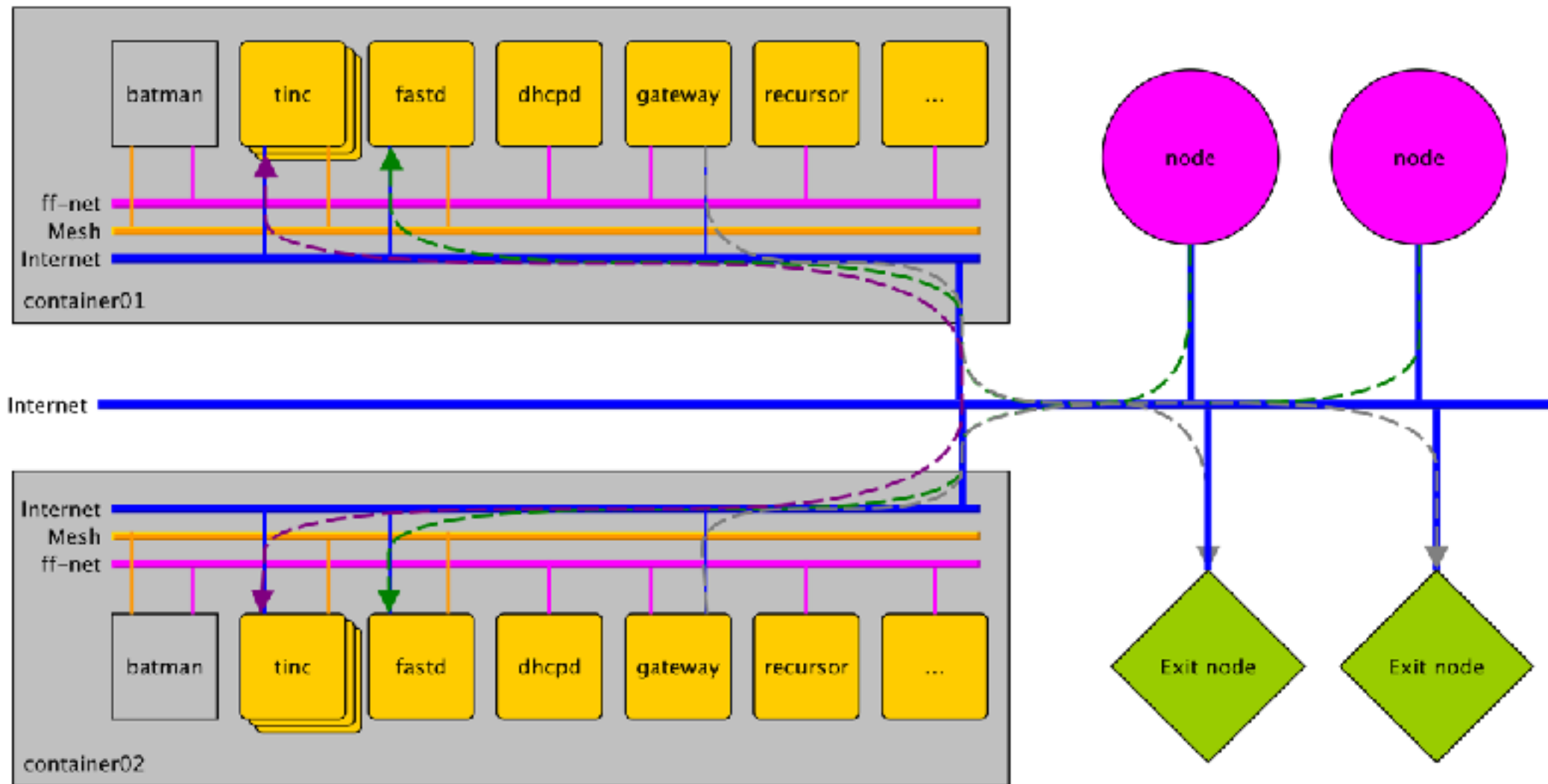
Docker...

## Virtual Machine



## Container

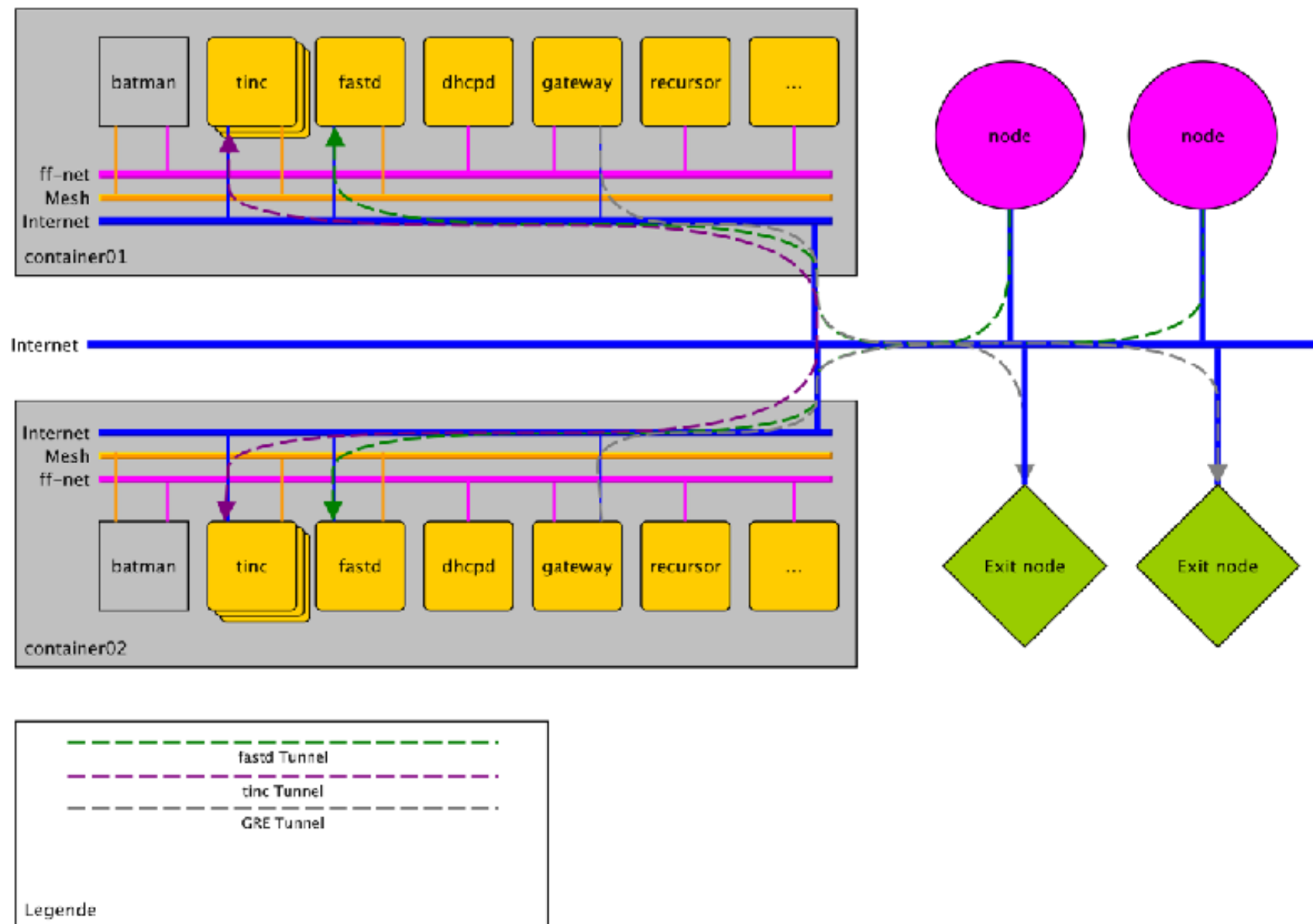




- Client verbindet sich an Node
- DHCP weist IP-Adresse & Gateway zu
- Client löst DNS-Namen beim Recursor auf
- Client schickt seine Pakete an Gateway
- Gateway nattet Pakete und schickt sie durchn den GRE-Tunnel zum Exit Node

# Freifunk in Containern

Alles schön eingepackt



## Docker-Images

- ffhef/debian-batman
- ffhef/tinc
- ffhef/fastd
- ffhef/dhcpd
- ffhef/gateway-ipv4
- ffhef/recursor

### **Konfiguration** › setup

- Installiert Docker und B.A.T.M.A.N. auf dem Server
- Legt Keys und Konfigurationen für Container an

### **Orchestrierung** › start

- Startet die Container
  - Konfiguration größtenteils über Umgebungsvariablen

```
#
# Freifunk configuration
#
freifunk_image_tag="latest"
freifunk_name=Freifunk Hennef
freifunk_shortname=ffhef

# Mesh
freifunk_mesh_ipv4_net=10.186.0.0/16
freifunk_mesh_mac_begin="fe:ed:ca:fe:"

# Directories
freifunk_container_data_dir=/freifunk

# Batman
#####
freifunk_batman_adv_version="2017.0.1"
freifunk_batctl_version="2017.0"
freifunk_batman_address_ipv4="{{ freifunk_mesh_ipv4_net | ipaddr( 10 + (freifunk_group_index|int) ) |
ipaddr('ip') }}"

# Tinc
#####
freifunk_tinc_name="{{ inventory_hostname_short }}"
freifunk_tinc_interface_mac="{{ freifunk_mesh_mac_begin }}0{{ freifunk_group_index|int }}:[01"

...
```



## State

- Alle notwendigen Dateien liegen in `/freifunk`
- Mit Ansible Inventory ausreichend für Betrieb
  - Einfaches Backup

## Konfiguration

- Wenn möglich über Umgebungsvariablen
- Template liegt im Image
- `envsubst` erstellt eigentliche Konfiguration

## › Beispiel

```
/freifunk/tinc/rsa_key.priv  
/freifunk/tinc/rsa_key.pub  
/freifunk/tinc/hosts  
/freifunk/tinc/hosts/container01  
/freifunk/tinc/hosts/container02  
  
/freifunk/fastd/fastd.public  
/freifunk/fastd/fastd.secret  
  
/freifunk/dhcpd/dhcpd.leases
```



## **Ansible** › ffhef.freifunk\_node

- Installiert Kernel-Modul & batctl auf dem Server
- Legt Docker-Network „freifunk-net“ an
  - Wird mit „bat0“ gebridged

## **Dockerimage** › ffhef/debian-batman

- Basisimage ffhef/debian-batman
- Tools
  - batctl, curl, envsubst, git...

## **Ansible** › ffhef.tinc

- Erzeugt Keys
- Kopiert Keys auf und von anderen Servern

## **Docker** › ffhef/tinc

- Verwendet Hostnetzwerk
- Verbindet sich mit allen anderen Freifunkservern
- Ermöglicht direkte Kommunikation im Mesh mit anderen Servern und deren Nodes & Clients

### **Ansible** › ffhef.fastd

- Erzeugt Keys
- Startet mehrere Instanzen  
`freifunk_fastd_instances="{{ ansible_processor_cores }}"`
- Legt iptables-Regeln zum Loadbalancing an  
`iptables ... -m statistic --mode nth --every 8`

### **Docker** › ffhef/tinc

- Verwendet Hostnetzwerk
- Verbindet sich mit allen anderen Freifunkservern
- Ermöglicht direkte Kommunikation im Mesh mit anderen Servern und deren Nodes & Clients

# DHCP & Recursors

Eine Adresse für jeden...

## **Ansible** › ffhef.dhcpd & ffhef.recursor

- Startet Container mit richtigen Umgebungsvariablen

## **Docker** › ffhef/dhcpd & ffhef/recursor

- DHCP: Verwendet Hostnetzwerk
- DHCP: Vergibt IP-Adressen
- Recursor: Löst IP-Adressen auf...

## **Ansible** › ffhef.gateway-ipv4

- Konfiguration für Server in `host_vars/<SERVER>`
  - GRE-Tunnel
  - BGP-Neighbors

## **Docker** › ffhef/gateway-ipv4

- Startet GRE-Tunnel
- iptables NAT-Regeln
- Führt BIRD aus und baut BGP-Sessions auf

# Minimalkonfiguration

In \$kurzerZeit zur eigenen Freifunk-Infrastruktur

## hosts

```
freifunk_name=Freifunk Hennef
freifunk_shortname=ffhef
freifunk_mesh_ipv4_net=10.186.0.0/16
```

## host\_vars/container01.freifunk-hennef.de

```
freifunk_gateway:
  peering_public_address_ipv4: 185.66.193.86
  mesh_address_ipv4: "{{ freifunk_mesh_ipv4_net | ipaddr( 256 + freifunk_group_index|int ) | ipaddr('ip') }}"
  gre_tunnels:
    - name: a_ak_ber
      tunnel_endpoint: 185.66.195.0
      tunnel_address_ipv4: 100.64.1.69
      tunnel_peer_ipv4: 100.64.1.68
    - name: a_ix_dus
      tunnel_endpoint: 185.66.193.0
      tunnel_address_ipv4: 100.64.1.71
      tunnel_peer_ipv4: 100.64.1.70
    - name: b_ak_ber
      tunnel_endpoint: 185.66.195.1
      tunnel_address_ipv4: 100.64.1.73
      tunnel_peer_ipv4: 100.64.1.72
    - name: b_ix_dus
      tunnel_endpoint: 185.66.193.1
      tunnel_address_ipv4: 100.64.1.75
      tunnel_peer_ipv4: 100.64.1.74
```

# Demo

SID: Freifunk-Hennef ;)

# Demo



- Dokumentation
- Community
- IPv6
- DNS-Server
- Backup
- Monitoring
  - Hopglass
  - Prometheus
  - Grafana

**Mattheus Happe**  
@tican

[Folgen](#)

Kein IPv6 auf dem #fftag17 @freifunk\_hennef

The screenshot shows a network connectivity test interface. On the left, under 'IPv4 connectivity', it lists: IPv4 (Supported), Address (193.58.193.87), Hostname (None), and ISP (Freifunk Rheinland Subnetwork). Below this, under 'IPv6 connectivity', it lists: IPv6 (Not supported), Address, Type, SLAAC, ICMP, Hostname, and ISP. On the right, under 'Score', it shows a progress bar at 4/20. Below that, under 'Browser', it lists: Default (IPv4) and Fallback (No). Under 'DNS', it lists: DNS4 + IP4 (Unresolvable), DNS6 + IP4 (Resolvable), and DNS6 + IP6 (Unresolvable). At the bottom right, there are buttons for 'Speed test' and 'Ping test'. The timestamp at the bottom reads '02:17 - 1. Apr. 2017 aus Rhein-Sieg-Kreis, Nordrhein-Westfalen'.

# Fragen & Diskussion



[github.com/Freifunk-Hennef](https://github.com/Freifunk-Hennef)



# Container?

Kurz mal reingucken...

## Demo