From Pi to Cluster

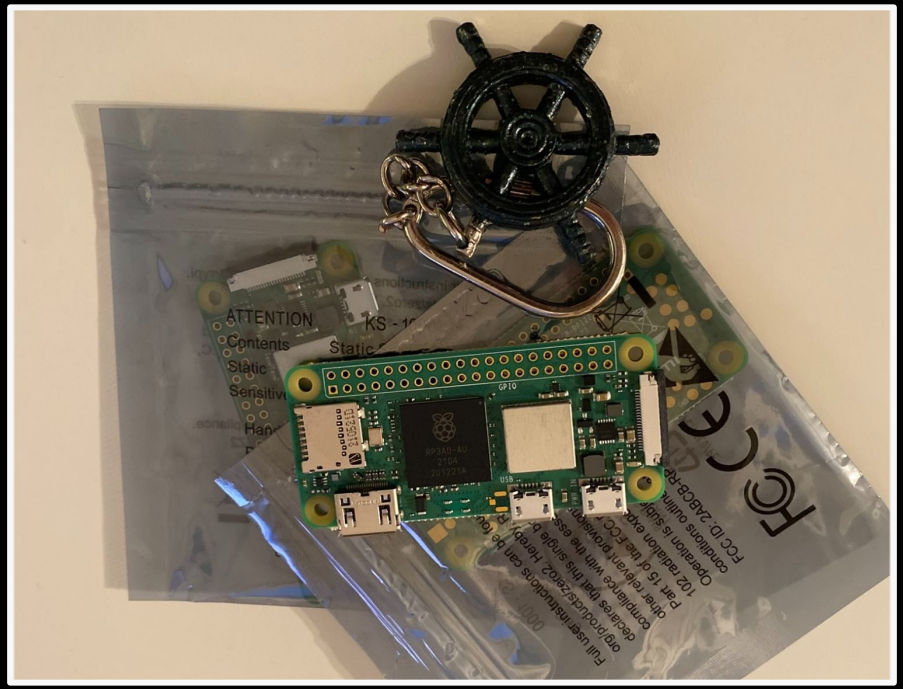
A Tale of Clusters and Chaos

Verena Traub

techcamp Hamburg 2025

techcamp HAMBURG

Former **HR Manager & Tech Recruiter**
turned into **Web Developer** (Bootcamp)
turned into **AWS Cloud Consultant**
turned into Cloud & DevOps Engineer - and starting with **Kubernetes** in 2024


ATTENTION
Contents
Static
GPIO
USB


YOU GET A CLUSTER! AND YOU GET A CLUSTER!
EVERYONE GETS A CLUSTER!
imgflip.com

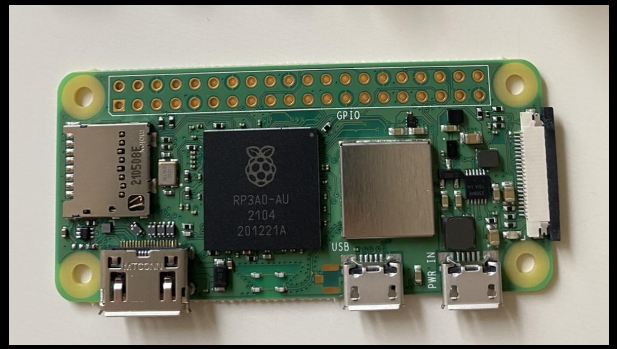
The Plan

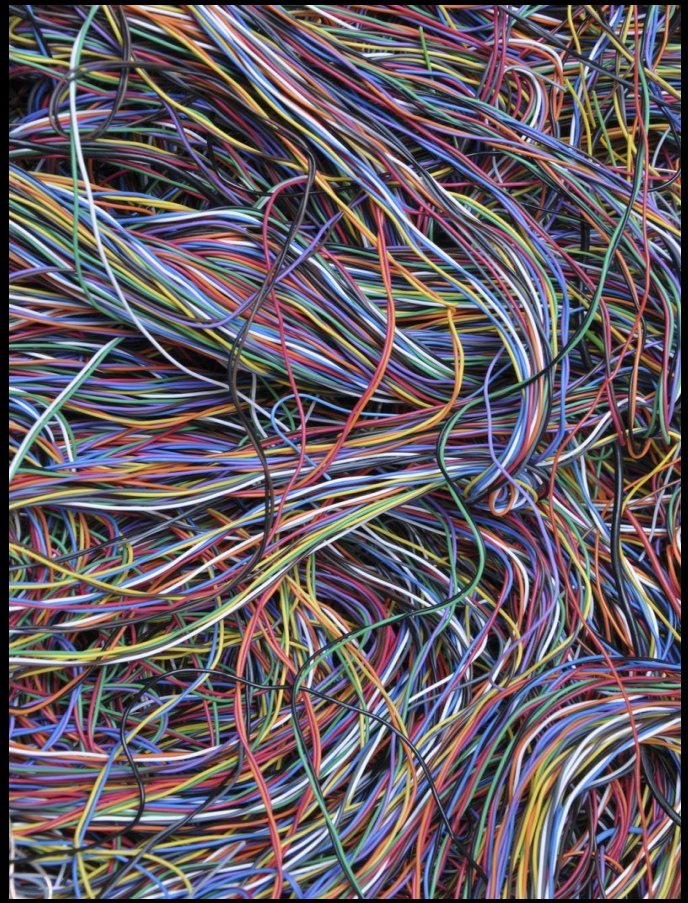
- Understand how to get rollin’
- Install k3s and deploy a test application
- Build a cluster (WiFi enabled)
- Build a cluster (Ethernet only)
- Build hybrid setup

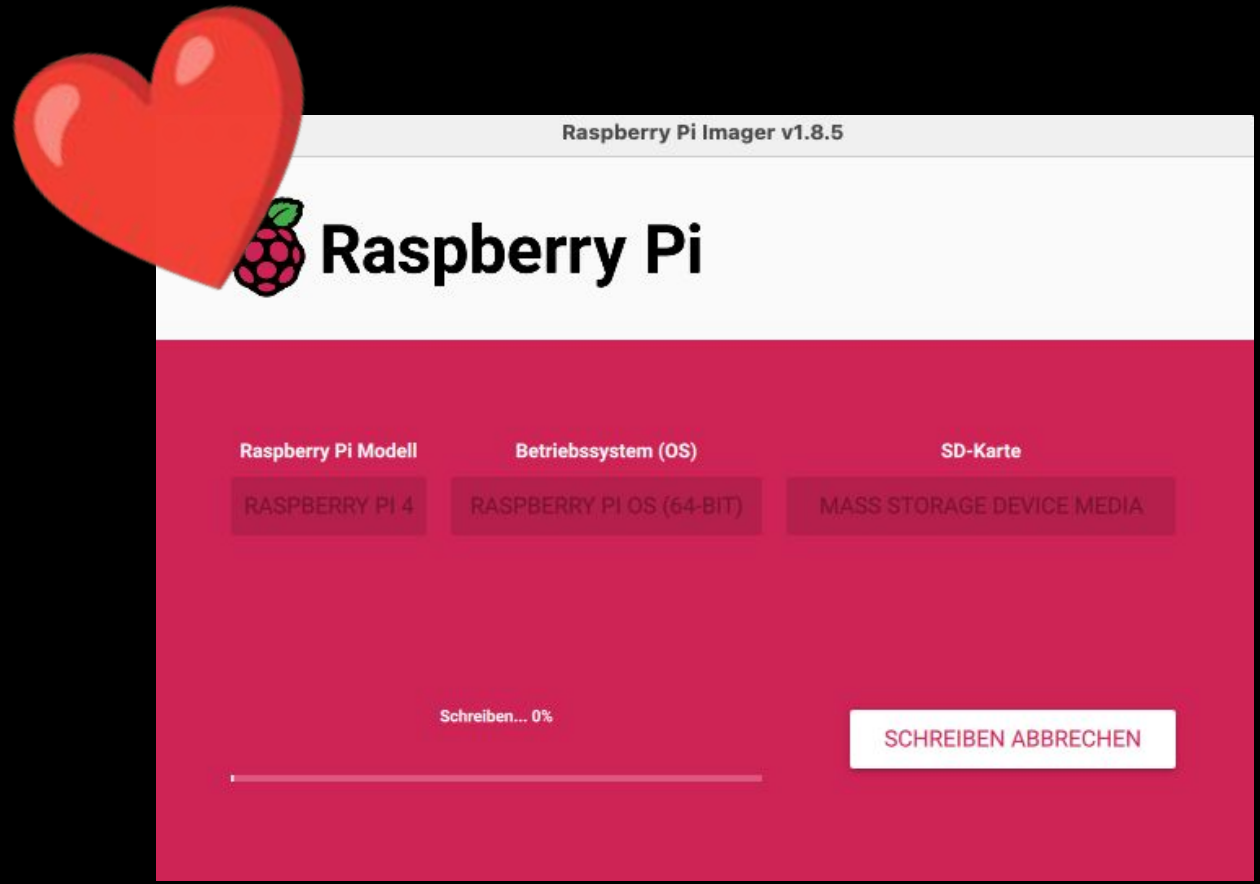
The Plan

- **Understand how to get rollin'**
- Install k3s and deploy a test application
- Build a cluster (WiFi enabled)
- Build a cluster (Ethernet only)
- Build hybrid setup

What is needed






Raspberry Pi Imager v1.8.5
Raspberry Pi
Raspberry Pi Modell
Betriebssystem (OS)
SD-Karte
RASPBERRY PI 4
RASPBERRY PI OS (64-BIT)
MASS STORAGE DEVICE MEDIA
Schreiben... 0%
SCHREIBEN ABBRECHEN

```
ssh verenatraub@raspberrypi.local
```

```
Linux raspberrypi 6.12.25+rpt-rpi-v8 #1 SMP PREEMPT Debian 1:6.12.25-1+rpt1 (2025-04-30) aarch64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue May 13 02:18:06 2025
verenatraub@raspberrypi:~ $
```

The Plan

- Understand how to get rollin' ✓
- **Install k3s and deploy a test application**
- Build a cluster (WiFi enabled)
- Build a cluster (Ethernet only)
- Build hybrid setup


```
sudo dphys-swapfile swapoff

sudo systemctl disable dphys-swapfile
```


```
curl -sfL https://get.k3s.io | sh -
```


```
curl -sfL https://get.k3s.io | sh -
```

```
Job for k3s.service failed because the control process exited with error code.
See "systemctl status k3s.service" and "journalctl -xeu k3s.service" for details.
verenatraub@raspberrypi:~ $ systemctl status k3s.service
● k3s.service - Lightweight Kubernetes
     Loaded: loaded (/etc/systemd/system/k3s.service; enabled; preset: enabled)
     Active: activating (auto-restart) (Result: exit-code) since Fri 2025-05-23 10:44:53 CEST; 3s ago
       Docs: https://k3s.io
    Process: 13637 ExecStartPre=/bin/sh -xc ! /usr/bin/systemctl is-enabled --quiet nm-cloud-setup.service 2>/dev/null (code=exited, status=0/SUCCESS)
    Process: 13639 ExecStartPre=/sbin/modprobe br_netfilter (code=exited, status=0/SUCCESS)
    Process: 13640 ExecStartPre=/sbin/modprobe overlay (code=exited, status=0/SUCCESS)
    Process: 13641 ExecStart=/usr/local/bin/k3s server (code=exited, status=1/FAILURE)
   Main PID: 13641 (code=exited, status=1/FAILURE)
        CPU: 981ms
```

b'

/boot/firmware/cmdline.txt

```
systemd.unified_cgroup_hierarchy=1 cgroup_enable=memory cgroup_memory=1
```


```
verenatraub@raspberrypi:~ $ sudo kubectl get nodes
NAME          STATUS   ROLES                  AGE   VERSION
raspberrypi   Ready    control-plane,master   83s   v1.32.4+k3s1
```


```
sudo kubectl create deployment nginx --image=nginx

sudo kubectl expose deployment nginx --port =80 --type=NodePort
```

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

b'

The Plan

- Understand how to get rollin' ✓
- Install k3s and deploy a test application ✓
- **Build a cluster (WiFi enabled)**
- Build a cluster (Ethernet only)
- Build hybrid setup

On the control plane to be:

```
sudo cat /var/lib/rancher/k3s/server/node-token
```

On the (soon to be) worker nodes:

```
curl -sfL https://get.k3s.io | K3S_URL=https://<CONTROL PLANE IP>:6443 \
K3S TOKEN=<MY NODE TOKEN> \
sh -
```

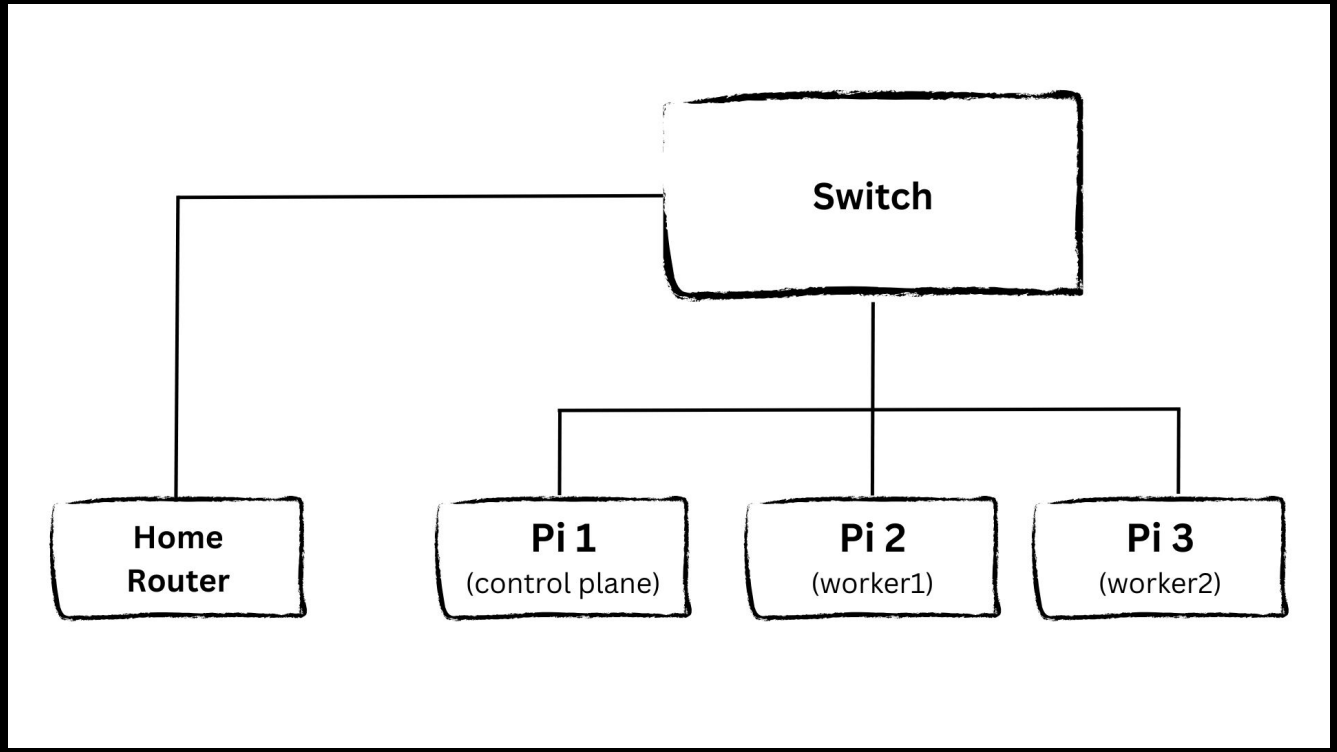
```
verenatraub@cp:~ $ sudo kubectl get nodes
NAME       STATUS   ROLES
cp         Ready    control-plane,master
worker1    Ready    <none>
worker2    Ready    <none>
```

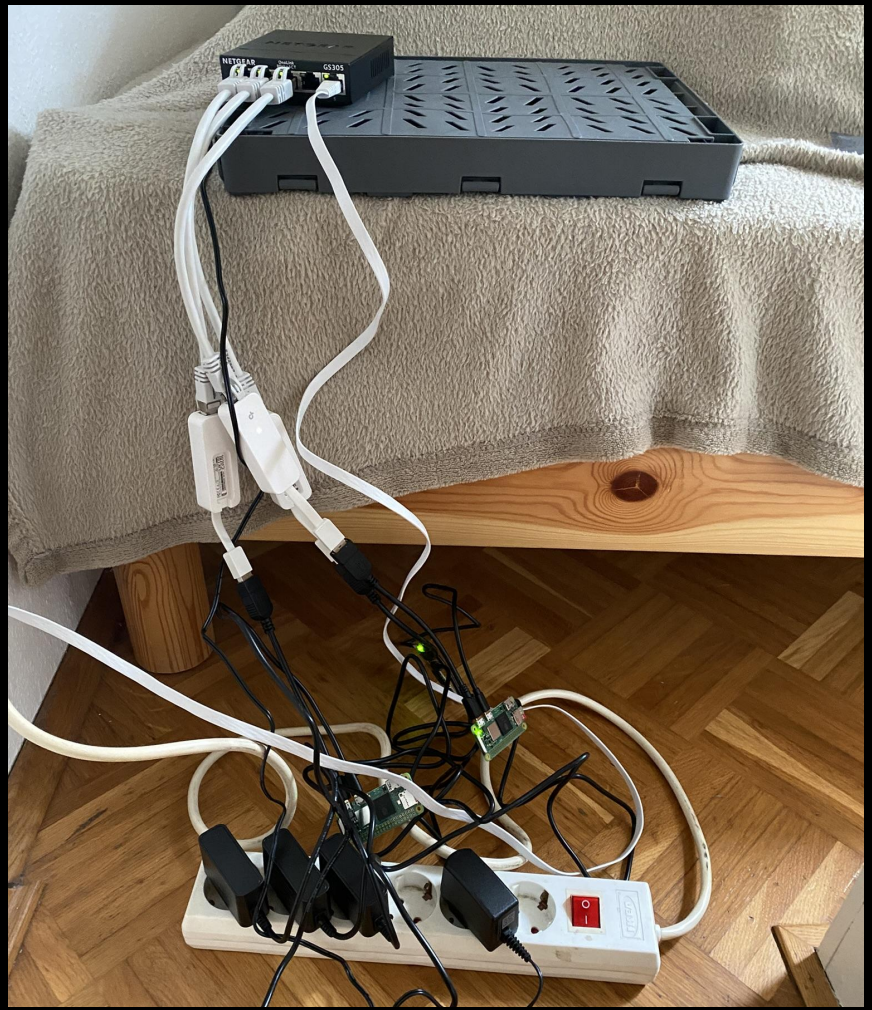
```
sudo kubectl taint nodes cp
node-role.kubernetes.io/control-plane =:NoSchedule
```


Well that was easy

The Plan

- Understand how to get rollin' ✓
- Install k3s and deploy a test application ✓
- Build a cluster (WiFi enabled) ✓
- **Build a cluster (Ethernet only)**
- Build hybrid setup

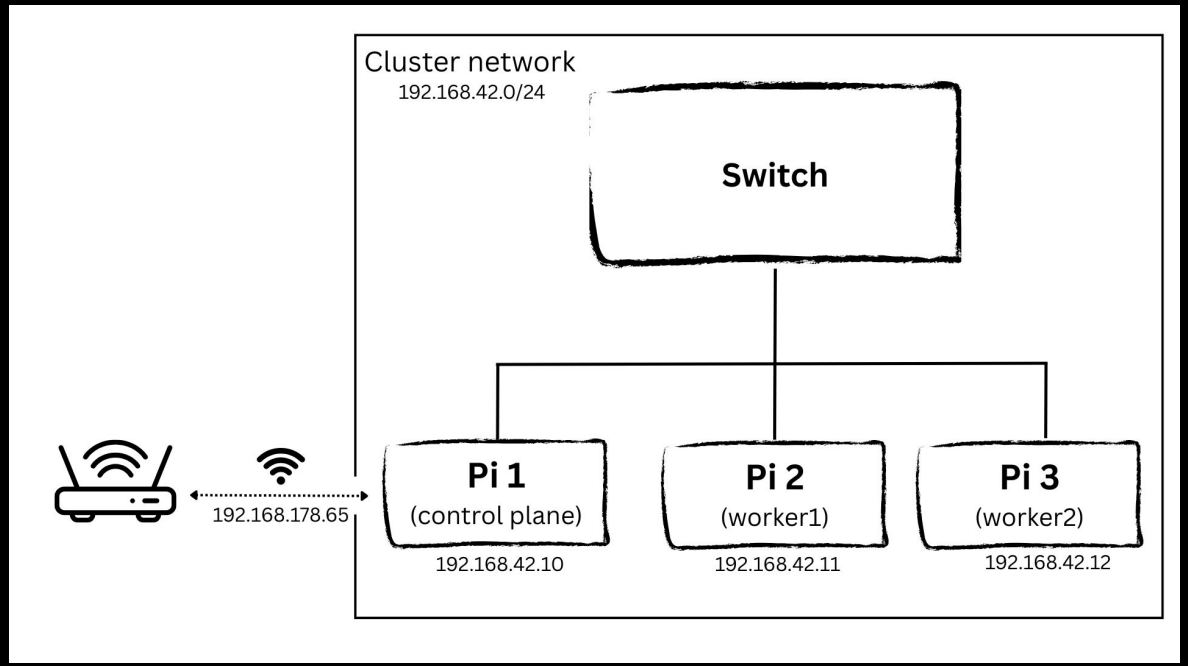

Switch
Home Router
Pi 1
(control plane)
Pi 2
(worker1)
Pi 3
(worker2)


NETGEAR
GS305

```
cannot resolve cp.local: Unknown host
```

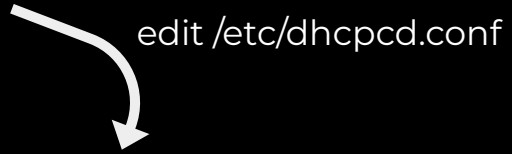
The Plan

- Understand how to get rollin' ✓
- Install k3s and deploy a test application ✓
- Build a cluster (WiFi enabled) ✓
- ~~Build a cluster (Ethernet only)~~ 🤬
- **Build hybrid setup**


Cluster network
192.168.42.0/24
Switch
192.168.178.65
Pi 1
(control plane)
192.168.42.10
Pi 2
(worker1)
192.168.42.11
Pi 3
(worker2)
192.168.42.12

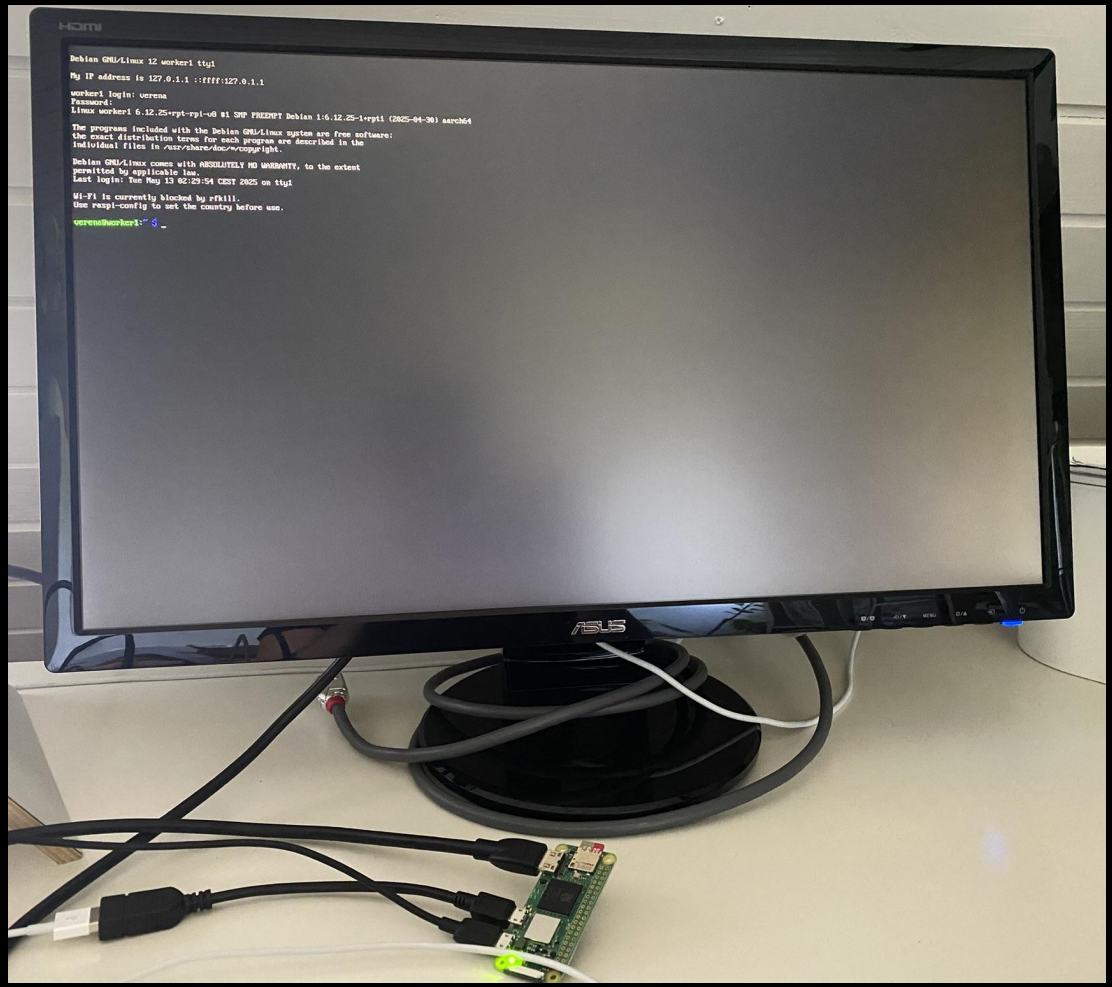
On the control plane

```
sudo apt update
sudo apt install dhcpcd5 -y
sudo systemctl enable dhcpcd
sudo systemctl start dhcpcd
```




```
interface eth0
static ip_address=192.168.42.10/24
```


```
ip addr show eth0
```


HDMI
Debian GNU/Linux 12 worker1 tty1
My IP address is 127.0.1.1 ::ffff:127.0.1.1
worker1 login: verena
Password:
Linux worker1 6.12.25+rpt-rpi-v8 #1 SMP PREEMPT Debian 1:6.12.25-1+rpt1 (2025-04-30) aarch64
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue May 13 02:29:54 CEST 2025 on tty1
Wi-Fi is currently blocked by rfkill.
Use raspi-config to set the country before use.
verena@worker1:~ $
ASUS

On the worker nodes

/etc/systemd/network/10-eth0.network

```
[Match]
Name=eth0

[Network]
Address=192.168.42.11/24 # 12 for
worker2
Gateway=192.168.42.10
```

```
sudo systemctl enable systemd-networkd
sudo systemctl start systemd-networkd
```

reboot

```
verena@cp:~ $ ssh verena@192.168.42.11
verena@192.168.42.11's password:
Linux worker1 6.12.25+rpt-rpi-v8 #1 SMP PREEMPT Debian 1:6.12.25-1+rpt1 (2025-04-30) aarch64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue May 13 03:02:41 2025

Wi-Fi is currently blocked by rfkill.
Use raspi-config to set the country before use.

verena@worker1:~ $
```

BONUS: Update /etc/hosts

Back on track 🎉 Hello k3s again

... but ...

...

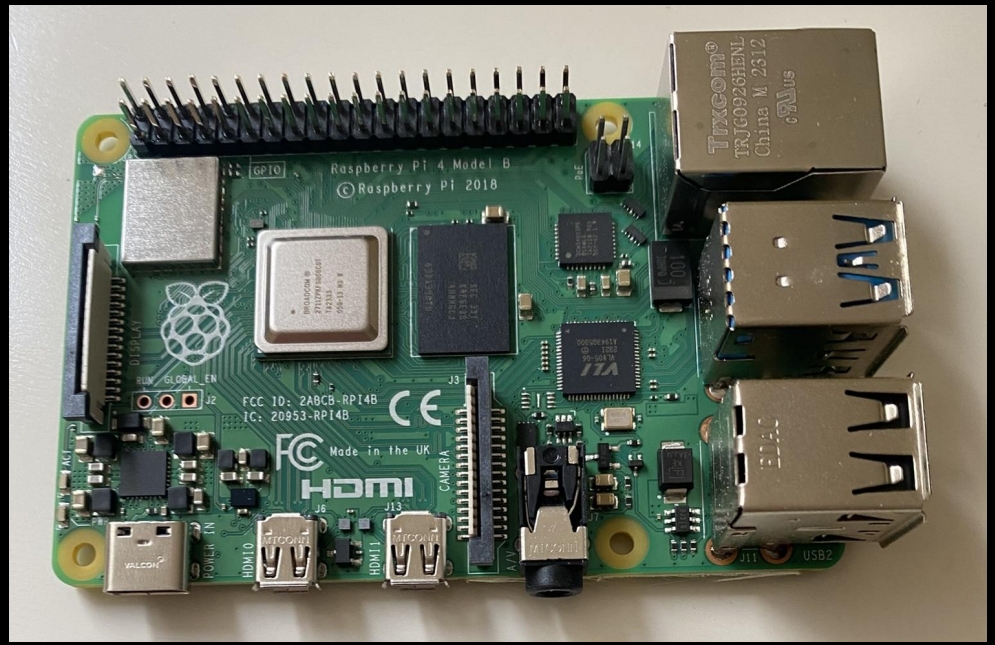
```
verena@cp:~ $ free -h
               total        used        free      shared  buff/cache   available
Mem:           416Mi       381Mi        37Mi       2.2Mi        47Mi        35Mi
Swap:            0B          0B          0B
```

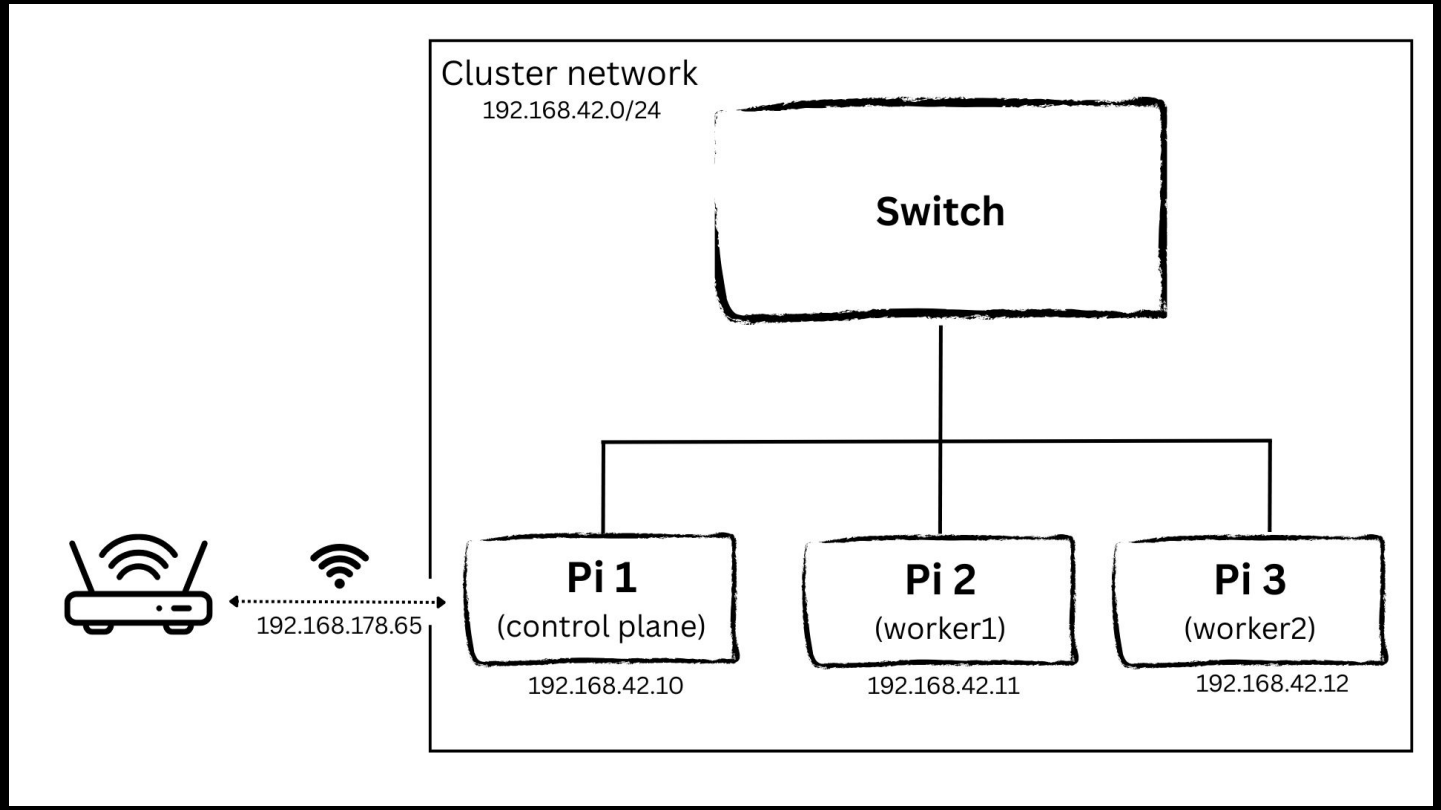

#FALLONTONIGHT
TOO MANY OPTIONS

Still ...

```
verena@cp:~ $ sudo kubectl taint nodes cp node-role.kubernetes.io/control-plane=:NoSchedule
Unable to connect to the server: net/http: TLS handshake timeout
```

Raspberry Pi 4 to the rescue




Cluster network
192.168.42.0/24
Switch
192.168.178.65
Pi 1
(control plane)
192.168.42.10
Pi 2
(worker1)
192.168.42.11
Pi 3
(worker2)
192.168.42.12

Now ... REALLY ... k3s 🎉

```
verena@cp:~ $ sudo kubectl get nodes
NAME   STATUS   ROLES                  AGE   VERSION
cp     Ready    control-plane,master   12s   v1.32.5+k3s1
verena@cp:~ $ free -h
               total        used        free      shared  buff/cache   available
Mem:           1.8Gi       558Mi       799Mi       4.8Mi       558Mi       1.3Gi
Swap:             0B          0B          0B
```

Make control plane also act as a NAT gateway

/etc/sysctl.conf

```
net.ipv4.ip_forward=1
```

```
sudo sysctl -p

sudo apt install iptables
sudo iptables -t nat -A POSTROUTING -o wlan0 -s
192.168.42.0/24 -j MASQUERADE

sudo apt install iptables-persistent
sudo netfilter-persistent save
```

Add domain_name_server
on workers and reboot all

Make control plane also a NAT gateway

/etc/sysctl.conf

```
net.ipv4.ip_forward=1
```

```
sudo sysctl -p

sudo apt install iptables
sudo iptables -t nat -A POSTROUTING -o wlan0 -s
192.168.42.0/24 -j MASQUERADE

sudo apt install iptables-persistent
sudo netfilter-persistent save
```

Add domain_name_server
on workers and reboot all

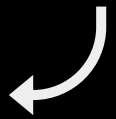
Make control plane also a NAT gateway

```
sudo nano /etc/systemd/system/nat-rules.service
```

```
[Unit]
Description=Set NAT iptables rule
After=network-online.target
Wants=network-online.target

[Service]
ExecStart=/sbin/iptables -t nat -A POSTROUTING -o
wlan0 -s 192.168.42.0/24 -j MASQUERADE
Type=oneshot
RemainAfterExit=yes

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



```
sudo systemctl daemon-reexec
sudo systemctl daemon-reload
sudo systemctl enable nat-rules.service
```

Make control plane also a NAT gateway

```
sudo nano /etc/systemd/system/nat-rules.service
```

```
[Unit]
Description=Set NAT iptables rule
After=network-online.target
Wants=network-online.target

[Service]
ExecStart=/sbin/iptables -t nat -A POSTROUTING -o
wlan0 -s 192.168.42.0/24 -j MASQUERADE
Type=oneshot
RemainAfterExit=yes

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

```
sudo systemctl daemon-reexec
sudo systemctl daemon-reload
sudo systemctl enable nat-rules.service
```

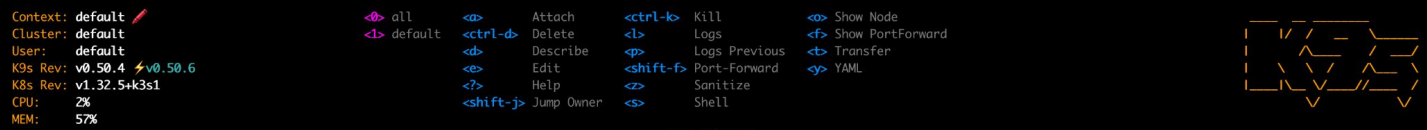
Now ... REALLY ... k3s 🎉

```
verena@worker1:~ $ curl -sfL https://get.k3s.io | K3S_URL=https://192.168.42.10:6443 \
K3S_TOKEN=K10d830246674667d626e5edbffb8fbbcf7308086bc6e65e8bd7dcb5586c1b8609d::server:4267e2c2440c5b36812f99e5f6367e9a \
 sh -
[INFO]  Finding release for channel stable
[INFO]  Using v1.32.5+k3s1 as release
[INFO]  Downloading hash https://github.com/k3s-io/k3s/releases/download/v1.32.5+k3s1/sha256sum-arm64.txt
[INFO]  Downloading binary https://github.com/k3s-io/k3s/releases/download/v1.32.5+k3s1/k3s-arm64
[INFO]  Verifying binary download
[INFO]  Installing k3s to /usr/local/bin/k3s
[INFO]  Skipping installation of SELinux RPM
[INFO]  Creating /usr/local/bin/kubectl symlink to k3s
[INFO]  Creating /usr/local/bin/crictl symlink to k3s
[INFO]  Creating /usr/local/bin/ctr symlink to k3s
[INFO]  Creating killall script /usr/local/bin/k3s-killall.sh
[INFO]  Creating uninstall script /usr/local/bin/k3s-agent-uninstall.sh
[INFO]  env: Creating environment file /etc/systemd/system/k3s-agent.service.env
[INFO]  systemd: Creating service file /etc/systemd/system/k3s-agent.service
[INFO]  systemd: Enabling k3s-agent unit
Created symlink /etc/systemd/system/multi-user.target.wants/k3s-agent.service → /etc/systemd/system/k3s-agent.service.
[INFO]  Host iptables-save/iptables-restore tools not found
[INFO]  Host ip6tables-save/ip6tables-restore tools not found
[INFO]  systemd: Starting k3s-agent
```

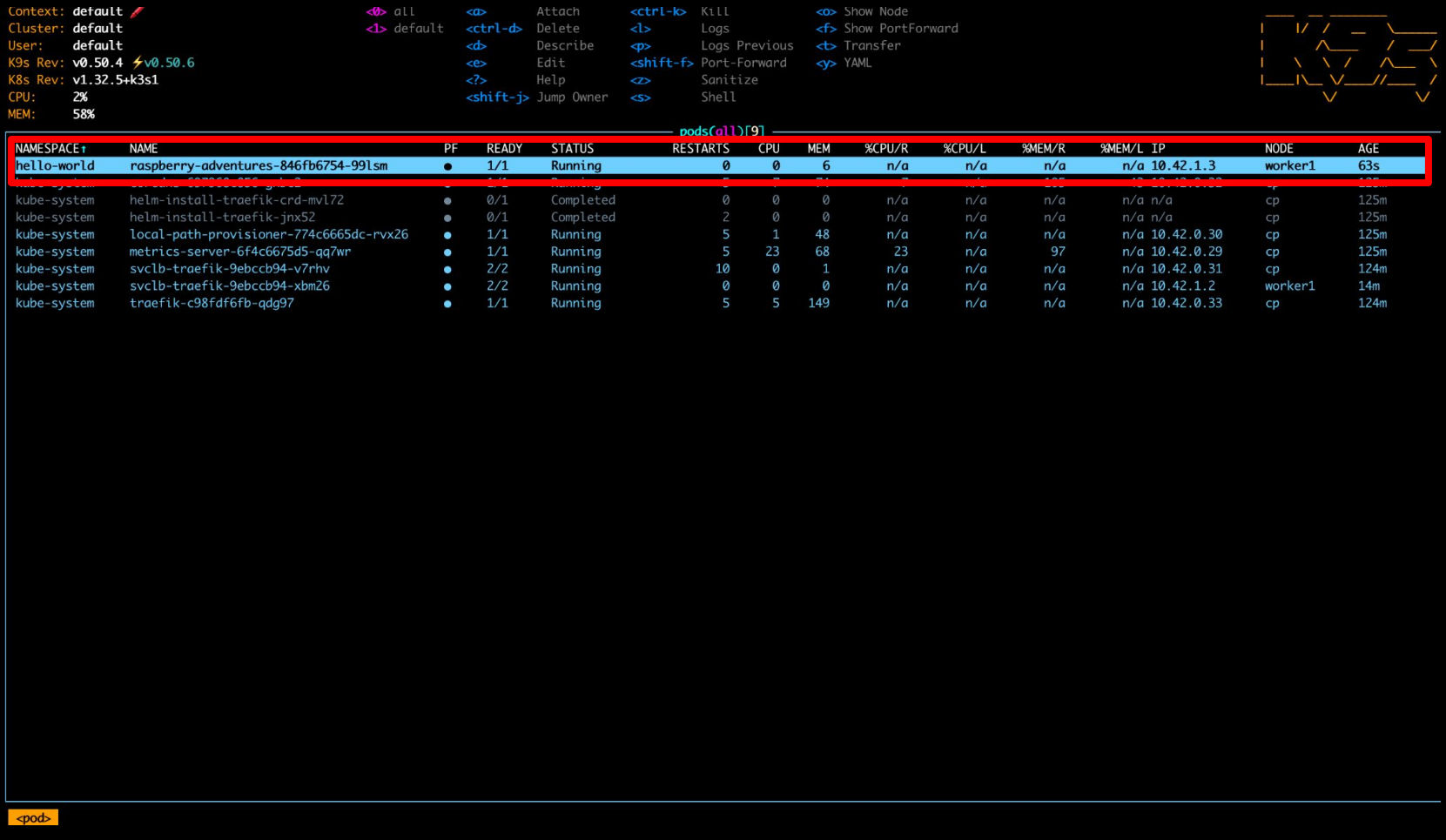
And FINALLY 🎉

```
verena@cp:~ $ sudo kubectl get nodes
NAME      STATUS   ROLES                  AGE     VERSION
cp        Ready    control-plane,master   118m    v1.32.5+k3s1
worker1   Ready    <none>                 6m44s   v1.32.5+k3s1
```

```
sudo cat /etc/rancher/k3s/k3s.yaml
```



NAMESPACE↑	NAME	PF	READY	STATUS	RESTARTS	CPU	MEM	%CPU/R	%CPU/L	%MEM/R	%MEM/L	IP	NODE	AGE
kube-system	coredns-697968c856-gnbc2	●	1/1	Running	5	7	73	7	n/a	105	43	10.42.0.32	cp	121m
kube-system	helm-install-traefik-crd-mvl72	●	0/1	Completed	0	0	0	n/a	n/a	n/a	n/a	n/a	cp	121m
kube-system	helm-install-traefik-jnx52	●	0/1	Completed	2	0	0	n/a	n/a	n/a	n/a	n/a	cp	121m
kube-system	local-path-provisioner-774c6665dc-rvx26	●	1/1	Running	5	1	48	n/a	n/a	n/a	n/a	10.42.0.30	cp	121m
kube-system	metrics-server-6f4c6675d5-qq7wr	●	1/1	Running	5	23	68	23	n/a	97	n/a	10.42.0.29	cp	121m
kube-system	svclb-traefik-9ebccb94-v7rhv	●	2/2	Running	10	0	1	n/a	n/a	n/a	n/a	10.42.0.31	cp	120m
kube-system	svclb-traefik-9ebccb94-xbm26	●	2/2	Running	0	0	0	n/a	n/a	n/a	n/a	10.42.1.2	worker1	10m
kube-system	traefik-c98fdf6fb-qdg97	●	1/1	Running	5	1	149	n/a	n/a	n/a	n/a	10.42.0.33	cp	120m


Context: default
Cluster: default
User: default
K9s Rev: v0.50.4 ⚡v0.50.6
K8s Rev: v1.32.5+k3s1
CPU: 2%
MEM: 58%
<0> all
<1> default
<a> Attach
<ctrl-d> Delete
<d> Describe
<e> Edit
<?> Help
<shift-j> Jump Owner
<ctrl-k> Kill
<l> Logs
<p> Logs Previous
<shift-f> Port-Forward
<z> Sanitize
<s> Shell
<o> Show Node
<f> Show PortForward
<t> Transfer
<y> YAML
pods(all)[9]
NAMESPACE↑ NAME PF READY STATUS RESTARTS CPU MEM %CPU/R %CPU/L %MEM/R %MEM/L IP NODE AGE
hello-world raspberry-adventures-846fb6754-99lsm 1/1 Running 0 0 6 n/a n/a n/a n/a 10.42.1.3 worker1 63s
kube-system helm-install-traefik-crd-mvl72 0/1 Completed 0 0 0 n/a n/a n/a n/a n/a cp 125m
kube-system helm-install-traefik-jnx52 0/1 Completed 2 0 0 n/a n/a n/a n/a n/a cp 125m
kube-system local-path-provisioner-774c6665dc-rvx26 1/1 Running 5 1 48 n/a n/a n/a n/a 10.42.0.30 cp 125m
kube-system metrics-server-6f4c6675d5-qq7wr 1/1 Running 5 23 68 23 n/a 97 n/a 10.42.0.29 cp 125m
kube-system svclb-traefik-9ebccb94-v7rhv 2/2 Running 10 0 1 n/a n/a n/a n/a 10.42.0.31 cp 124m
kube-system svclb-traefik-9ebccb94-xbm26 2/2 Running 0 0 0 n/a n/a n/a n/a 10.42.1.2 worker1 14m
kube-system traefik-c98fdf6fb-qdg97 1/1 Running 5 5 149 n/a n/a n/a n/a 10.42.0.33 cp 124m
<pod>

Nicht sicher raspberry-adventures.local

Zum Aktualisieren neu starten

03_ClientsUnsorte... 00_Taskboard-Tec... 00_Taskboard-La... 01_Taskboard-Org... Task Board - VT 01_Taskboard-Talk... Task Board - Dash... Alle Lesezeichen

Home

Verena Traub

Cloud Consulting & DevOps

Raspberry Adventures

It all started with my 40th birthday when I was gifted 3 Raspberry Pis. With some months of k8s experience I thought the same as quite likely anyone:

I should totally build my own cluster!


AND NOW
TIME FOR A LIVE DEMO


Demo
Time!
meme-arsenal.ru

What we can learn out of the demo disaster - and my adventure in general?

✅ Always, I mean really **ALWAYS** (!!!) have one (or two, or three, ...) spare SD cards!

✅ Cables are ... essential ... Especially the right ones 😇

✅ ChatGPT is not always your friend. But neither are blogs 🤷‍♀️

✅ Don’t give up despite the urge to do so 💪

Thanks for listening 🙏

Verena Traub

Verena Traub

verena@bnerd.com