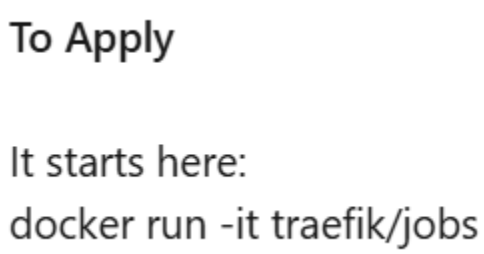
Traefik job

There is this new job available on LinkedIn: Platform Engineer @ Trafik Labs. That sounds interesting.

But what really caught my eye in the job description, all the way at the bottom, is this:



I am curious. What happens if I do that?

I have several Linux hosts at home with Docker configured, so let's try it out.

```
$ docker run -it traefik/jobs
K8s, where are you?
```

Oh, it seems it needs Kubernetes.

I set up a Kubernetes cluster using Talos, so let's give it a try. A simple pod should do.

> I tried setting up a normal pod with some restrictions, but I deployed a Talos cluster, and it has the pod security admission controller enabled by default on all namespaces (see: https://www.talos.dev/v1.10/kubernetes-guides/configuration/pod-security/). The pod indicates that it wants to run as root, so I had to set runAsNonRoot to false. But even then I got warnings. Since this is a test cluster, I updated the default namespace to not enforce the policies:
>
> ```
> $ kubectl label namespace default pod-security.kubernetes.io/enforce=privileged
> ```

```
$ cat pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: traefik-jobs
spec:
  containers:
  - name: traefik-jobs
    image: traefik/jobs:latest
    securityContext:
      privileged: false
      allowPrivilegeEscalation: false
      capabilities:
        drop: ["ALL"]
      runAsNonRoot: false
      seccompProfile:
        type: RuntimeDefault

$ kubectl apply -f pod.yaml
pod/traefik-jobs created

$ kubectl logs traefik-jobs
It seems I do need more permissions... May I be promoted cluster-admin?
Hmmmm, it seems the Deployment has an issue
```

Ah okay. I'm getting that this is becoming like a test - can you solve it? And if so, you're probably good enough to apply at Traefik Labs.

That's a challenge I'd like to take.

This is easy stuff; turn it into a deployment and give it cluster-admin access. A bit concerned about promoting the pod to cluster-admin, but hey, it's a test cluster on Talos that I set up anyway, so even if it messes with the cluster... I can always deploy a new one.

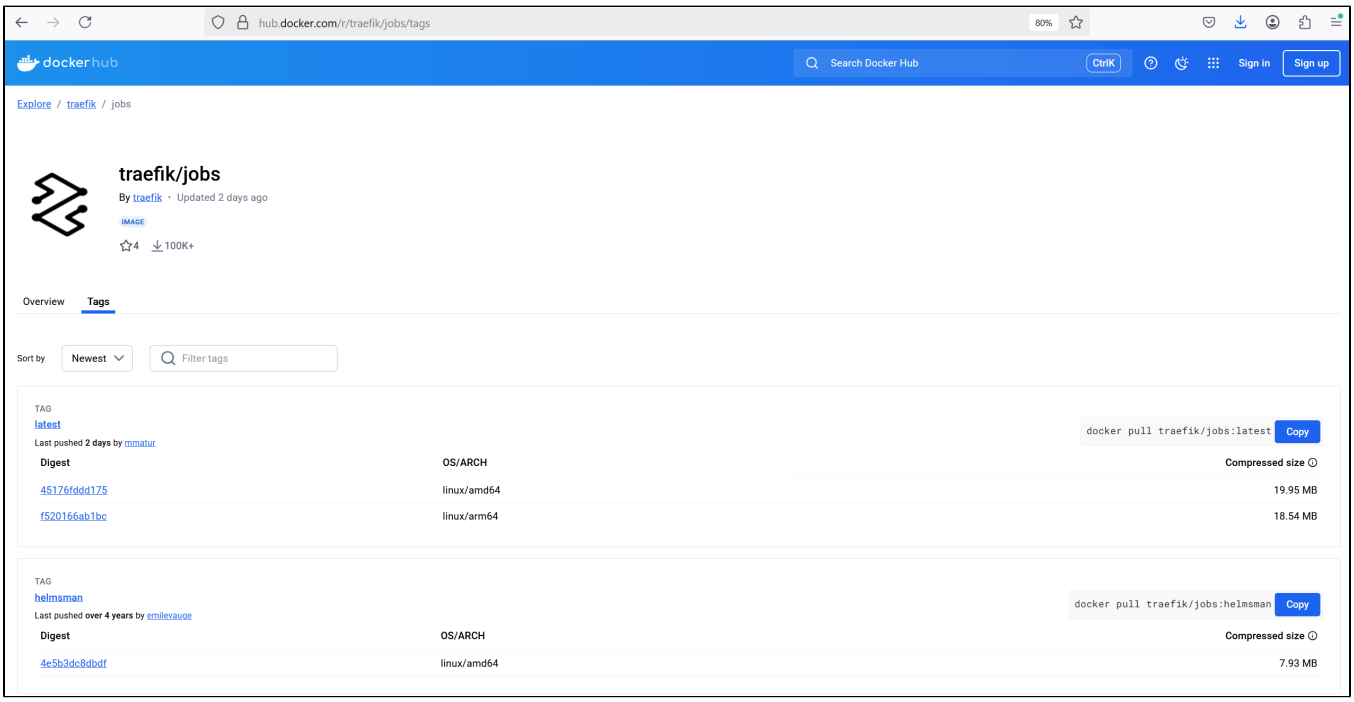
First, let's investigate the image.


```
$ docker image inspect docker.io/traefik/jobs
[
     {
          "Id": "588fd9d97e485e4e5df0fe53fc806c8a06b6a26c0c3d75d0224986a511de9c1c",
          "Digest": "sha256:45176fddd1752bb0f5fa0c64f5d7c8db0efd3decea00b689ca28a594a8970f6e",
          "RepoTags": [
               "docker.io/traefik/jobs:latest"
          ],
          "RepoDigests": [
               "docker.io/traefik/jobs@sha256:45176fddd1752bb0f5fa0c64f5d7c8db0efd3decea00b689ca28a594a8970f6e",
               "docker.io/traefik/jobs@sha256:86c14a6e0b138cd378a30b5bd7a685733b7ff7a5ecb8f5242a4f6ab7aa05e6a6"
          ],
          "Parent": "",
          "Comment": "buildkit.dockerfile.v0",
          "Created": "2025-05-16T08:12:06.422264617Z",
          "Config": {
               "Env": [
                    "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
               ],
               "Entrypoint": [
                    "/start"
               ],
               "WorkingDir": "/",
               "Labels": {
                    "traefik": "dcc9c530767c102764d45d621fc92317"
               }
          },
          "Version": "",
          "Author": "",
          "Architecture": "amd64",
          "Os": "linux",
          "Size": 46625248,
          "VirtualSize": 46625248,
          "GraphDriver": {
               "Name": "overlay",
               "Data": {
                    "UpperDir": "/home/meilinkm/.local/share/containers/storage/overlay
/f549adf2a127a31583f31d7fd3dfb0227c27ecb45518362cc0a74db931faf84b/diff",
                    "WorkDir": "/home/meilinkm/.local/share/containers/storage/overlay
/f549adf2a127a31583f31d7fd3dfb0227c27ecb45518362cc0a74db931faf84b/work"
               }
          },
          "RootFS": {
               "Type": "layers",
               "Layers": [
                    "sha256:f549adf2a127a31583f31d7fd3dfb0227c27ecb45518362cc0a74db931faf84b"
               ]
          },
          "Labels": {
               "traefik": "dcc9c530767c102764d45d621fc92317"
          },
          "Annotations": {},
          "ManifestType": "application/vnd.oci.image.manifest.v1+json",
          "User": "",
          "History": [
               {
                    "created": "2025-05-16T08:12:06.422264617Z",
                    "created_by": "LABEL traefik=dcc9c530767c102764d45d621fc92317",
                    "comment": "buildkit.dockerfile.v0",
                    "empty_layer": true
               },
               {
```

```
                "created": "2025-05-16T08:12:06.422264617Z",
                "created_by": "ARG TARGETPLATFORM",
                "comment": "buildkit.dockerfile.v0",
                "empty_layer": true
            },
            {
                "created": "2025-05-16T08:12:06.422264617Z",
                "created_by": "COPY ./dist/linux/amd64/helmsman /start # buildkit",
                "comment": "buildkit.dockerfile.v0"
            },
            {
                "created": "2025-05-16T08:12:06.422264617Z",
                "created_by": "ENTRYPOINT [\"/start\"]",
                "comment": "buildkit.dockerfile.v0",
                "empty_layer": true
            }
        ],
        "NamesHistory": [
            "docker.io/traefik/jobs:latest"
        ]
    }
]
```

Hmm... it looks like the image is om 2025-05-16. Rather new.

Indeed, it was recently updated:



Let's see what is inside the image:

```
$ docker save traefik/jobs > image.tar
$ ls
$ tar xvf f549adf2a127a31583f31d7fd3dfb0227c27ecb45518362cc0a74db931faf84b.tar
```

I see a file called "start". That's also the entrypoint for the Docker image.

I ran a strings command on it, and I can see the comments in it that it generates (like strings starting with "Hmmm".

```
$ strings start 2>&1 | sed "s/Hmm/\nHmm/g" | grep Hmmmm | cut -c1-60
Hmmmm, it seems K8s has an issue (>_<)crypto/sha512: invalid
Hmmmm, it seems the Deployment has an issue
Hmmmm, it seems I cannot get my hand to an IngressClass
```

You would think they would at least try to hide the strings by base64 encoding them, but no.

I tried searching for URLs in the strings output of the start binary, or tried searching for 'traefik' in it, to see if it would yield anything useful, but so far no.

The binary seems to have been built in Go:

```
$ file start
start: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, Go BuildID=Exn4KPFdd6d2PCv1pcDH
/V87aGChu6TVVvzhp_lz1/O1hcPsysFAEpNEdnsw5B/Q0NVP6PMuu_B_BAwcbtq, with debug_info, not stripped
```

Let's see what trivy thinks about any possible vulnerabilties in this image:

```
$ docker run aquasec/trivy image traefik/jobs:latest
Resolved "aquasec/trivy" as an alias (/home/meilinkm/.cache/containers/short-name-aliases.conf)
Trying to pull docker.io/aquasec/trivy:latest...
Getting image source signatures
Copying blob 21038c1e39ac done   |
Copying blob f18232174bc9 done   |
Copying blob ef7034495152 done   |
Copying blob 9cdae7569ace done   |
Copying config 89d1574daa done   |
Writing manifest to image destination
2025-05-18T17:32:40Z    INFO    [vulndb] Need to update DB
2025-05-18T17:32:40Z    INFO    [vulndb] Downloading vulnerability DB...
2025-05-18T17:32:40Z    INFO    [vulndb] Downloading artifact...        repo="mirror.gcr.io/aquasec/trivy-db:2"
16.00 MiB / 63.87 MiB [--------------->_____________________________________________] 25.05% ? p/s ?37.33 MiB /
63.87 MiB [---------------------------------->_____________________________] 58.45% ? p/s ?58.83 MiB / 63.87 MiB
[-------------------------------------------------------->____] 92.11% ? p/s ?63.87 MiB / 63.87 MiB
[----------------------------------------------->] 100.00% 79.73 MiB p/s ETA 0s63.87 MiB / 63.87 MiB
[----------------------------------------------->] 100.00% 79.73 MiB p/s ETA 0s63.87 MiB / 63.87 MiB
[----------------------------------------------->] 100.00% 79.73 MiB p/s ETA 0s63.87 MiB / 63.87 MiB
[----------------------------------------------->] 100.00% 74.58 MiB p/s ETA 0s63.87 MiB / 63.87 MiB
[----------------------------------------------->] 100.00% 74.58 MiB p/s ETA 0s63.87 MiB / 63.87 MiB
[----------------------------------------------->] 100.00% 74.58 MiB p/s ETA 0s63.87 MiB / 63.87 MiB
[----------------------------------------------->] 100.00% 69.77 MiB p/s ETA 0s63.87 MiB / 63.87 MiB
[----------------------------------------------->] 100.00% 69.77 MiB p/s ETA 0s63.87 MiB / 63.87 MiB
[----------------------------------------------->] 100.00% 69.77 MiB p/s ETA 0s63.87 MiB / 63.87 MiB
[----------------------------------------------->] 100.00% 65.27 MiB p/s ETA 0s63.87 MiB / 63.87 MiB
[----------------------------------------------->] 100.00% 65.27 MiB p/s ETA 0s63.87 MiB / 63.87 MiB
[-------------------------------------------------] 100.00% 23.17 MiB p/s 3.0s2025-05-18T17:32:43Z     INFO
[vulndb] Artifact successfully downloaded       repo="mirror.gcr.io/aquasec/trivy-db:2"
2025-05-18T17:32:43Z    INFO    [vuln] Vulnerability scanning is enabled
2025-05-18T17:32:43Z    INFO    [secret] Secret scanning is enabled
2025-05-18T17:32:43Z    INFO    [secret] If your scanning is slow, please try '--scanners vuln' to disable
secret scanning
2025-05-18T17:32:43Z    INFO    [secret] Please see also https://trivy.dev/v0.62/docs/scanner
/secret#recommendation for faster secret detection
2025-05-18T17:32:45Z    INFO    Number of language-specific files       num=1
2025-05-18T17:32:45Z    INFO    [gobinary] Detecting vulnerabilities...

Report Summary


 Target    Type    Vulnerabilities  Secrets

 start   gobinary         9            -

Legend:
- '-': Not scanned
- '0': Clean (no security findings detected)
```

```
start (gobinary)
================
Total: 9 (UNKNOWN: 0, LOW: 0, MEDIUM: 7, HIGH: 2, CRITICAL: 0)


          Library            Vulnerability   Severity  Status           Installed Version            Fixed
Version                             Title

 golang.org/x/net            CVE-2022-41723  HIGH      fixed   v0.3.1-0.20221206200815-1e63c2f08a10
0.7.0          golang.org/x/net/http2: avoid quadratic complexity in HPACK

                decoding

                https://avd.aquasec.com/nvd/cve-2022-41723

                            CVE-2023-39325
0.17.0         golang: net/http, x/net/http2: rapid stream resets can cause

                excessive work (CVE-2023-44487)

                https://avd.aquasec.com/nvd/cve-2023-39325

                            CVE-2022-41717  MEDIUM
0.4.0          golang: net/http: excessive memory growth in a Go server

                accepting HTTP/2 requests...

                https://avd.aquasec.com/nvd/cve-2022-41717

                            CVE-2023-3978
0.13.0         golang.org/x/net/html: Cross site scripting

                https://avd.aquasec.com/nvd/cve-2023-3978

                            CVE-2023-44487
0.17.0         HTTP/2: Multiple HTTP/2 enabled web servers are vulnerable

                to a DDoS attack...

                https://avd.aquasec.com/nvd/cve-2023-44487

                            CVE-2023-45288
0.23.0         golang: net/http, x/net/http2: unlimited number of

                CONTINUATION frames causes DoS

                https://avd.aquasec.com/nvd/cve-2023-45288

                            CVE-2025-22870
0.36.0         golang.org/x/net/proxy: golang.org/x/net/http/httpproxy:

                HTTP Proxy bypass using IPv6 Zone IDs in golang.org/x/net

                https://avd.aquasec.com/nvd/cve-2025-22870

                            CVE-2025-22872
0.38.0         golang.org/x/net/html: Incorrect Neutralization of Input

                During Web Page Generation in x/net in...

                https://avd.aquasec.com/nvd/cve-2025-22872

 google.golang.org/protobuf  CVE-2024-24786                   v1.28.1
1.33.0         golang-protobuf: encoding/protojson, internal/encoding/json:

                infinite loop in protojson.Unmarshal when unmarshaling

                certain forms of...

                https://avd.aquasec.com/nvd/cve-2024-24786
```

Okay, nothing too weird, just some versions that need updating.

Let's proceed with the deployment and the cluster-admin role:

```
$ cat deployment.yaml
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: traefik-job-svc
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: traefik-cluster-admin
subjects:
- kind: ServiceAccount
  name: traefik-job-svc
  namespace: default
roleRef:
  kind: ClusterRole
  name: cluster-admin
  apiGroup: rbac.authorization.k8s.io
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: traefik-jobs
  namespace: default
  labels:
    app: traefik-jobs
spec:
  selector:
    matchLabels:
      app: traefik-jobs
  template:
    metadata:
      labels:
        app: traefik-jobs
    spec:
      serviceAccountName: traefik-job-svc
      containers:
      - name: traefik-jobs
        image: traefik/jobs
        securityContext:
          privileged: false
          allowPrivilegeEscalation: false
          capabilities:
            drop: ["ALL"]
          runAsNonRoot: false
          seccompProfile:
            type: RuntimeDefault

$ kubectl delete -f pod.yaml
pod "traefik-jobs" deleted

$ kubectl apply -f deployment.yaml
serviceaccount/traefik-job-svc created
clusterrolebinding.rbac.authorization.k8s.io/traefik-cluster-admin created
deployment.apps/traefik-jobs created

$ kubectl get deploy
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
traefik-jobs   0/1     1            0           7s

$ kubectl get pods
NAME                            READY   STATUS   RESTARTS      AGE
traefik-jobs-7f5d8b6b78-kcvc7   0/1     Error    2 (20s ago)   23s

$ kubectl logs traefik-jobs-7f5d8b6b78-kcvc7
Look at me by the 8888 ingress
```

Oh, of course, I should have seen that one coming - they want an Ingress object! After all... it's from Traefik Labs. That's what they do.

Now here's the thing. I don't have an Ingress controller running in the cluster. I am not familiar with Traefik, so let's use Nginx Ingress Controller. See: Deploy Open Source Nginx.

Once that is deployed, I'll need a service object. Oh, and I'll need to tweek the deployment, so it uses port 8888. I'll need an ingress object as well.

```
$ cat deployment.yaml
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: traefik-job-svc
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: traefik-cluster-admin
subjects:
- kind: ServiceAccount
  name: traefik-job-svc
  namespace: default
roleRef:
  kind: ClusterRole
  name: cluster-admin
  apiGroup: rbac.authorization.k8s.io
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: traefik-jobs
  namespace: default
  labels:
    app: traefik-jobs
spec:
  selector:
    matchLabels:
      app: traefik-jobs
  template:
    metadata:
      labels:
        app: traefik-jobs
    spec:
      serviceAccountName: traefik-job-svc
      containers:
      - name: traefik-jobs
        image: traefik/jobs
        ports:
        - containerPort: 8888
        securityContext:
          privileged: false
          allowPrivilegeEscalation: false
          capabilities:
            drop: ["ALL"]
          runAsNonRoot: false
          seccompProfile:
            type: RuntimeDefault
---
apiVersion: v1
kind: Service
metadata:
  name: traefik-jobs-service
  namespace: default
spec:
  selector:
    app: traefik-jobs
  ports:
    - protocol: TCP
      port: 8888
      targetPort: 8888
---
apiVersion: networking.k8s.io/v1
```

```
kind: Ingress
metadata:
  name: traefik-jobs-ingress
  namespace: default
spec:
  ingressClassName: nginx
  rules:
  - host: traefik-jobs
    http:
      paths:
      - backend:
          service:
            name: traefik-jobs-service
            port:
              number: 8888
        path: /
        pathType: Prefix

$ kubectl get pod
NAME                            READY   STATUS             RESTARTS      AGE
traefik-jobs-547fb94b9c-gjkl9   0/1     CrashLoopBackOff   3 (36s ago)   87s

$ kubectl logs traefik-jobs-547fb94b9c-gjkl9
Come on, you are applying to Traefik Labs! Get yourself a decent ingress
```

LOL. Of course! I guess I'll have to use Treafik instead. I do tend to disagree with Traefik Labs here. Nginx is pretty decent ingress too.

I uninstalled the helm charts and deleted the namespaces for Nginx and MetalLB.

Install Traefic using Helm as described on https://doc.traefik.io/traefik/getting-started/install-traefik/#use-the-helm-chart.

Now check the pod - it is now in a running state, and its log says:

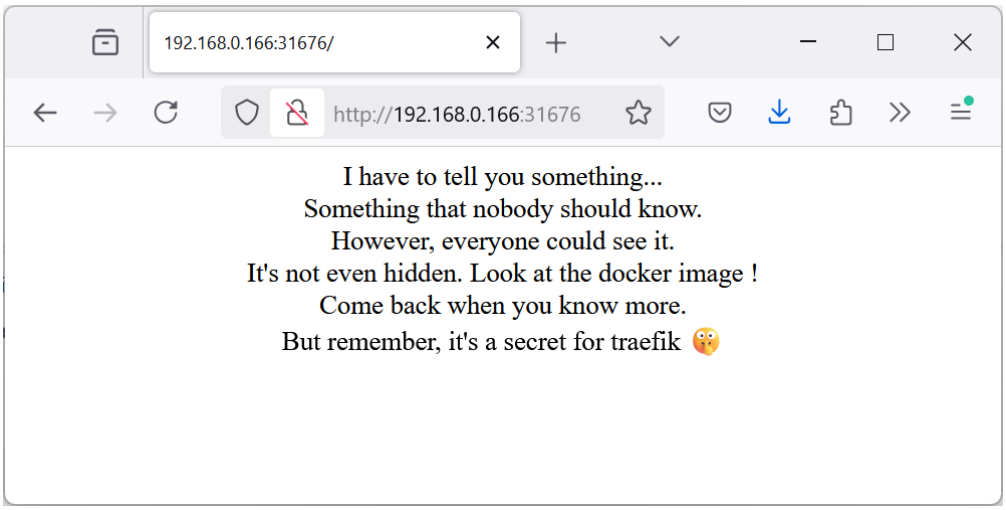
```
You have set up your cluster in good taste
Now that you have set up an ingress... You should be able find me...
```

I can see the ingress and the services:

```
$ kubectl get ingress
NAME                    CLASS     HOSTS   ADDRESS   PORTS   AGE
traefik-jobs-ingress    traefik   *                 80      3s

$ kubectl get svc -A
NAMESPACE     NAME                   TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)
AGE
default       kubernetes             ClusterIP      10.96.0.1       <none>        443/TCP
4d7h
default       traefik                LoadBalancer   10.109.93.208   <pending>     80:31676/TCP,443:32585/TCP
10m
default       traefik-jobs-service   ClusterIP      10.108.42.231   <none>        8888/TCP
8h
kube-system   kube-dns               ClusterIP      10.96.0.10      <none>        53/UDP,53/TCP,9153/TCP
4d7h
```

Hmmm. Treafik listens on port 31676; let's access that.



Ah. A riddle. Okay, so there is a secret in the docker image.

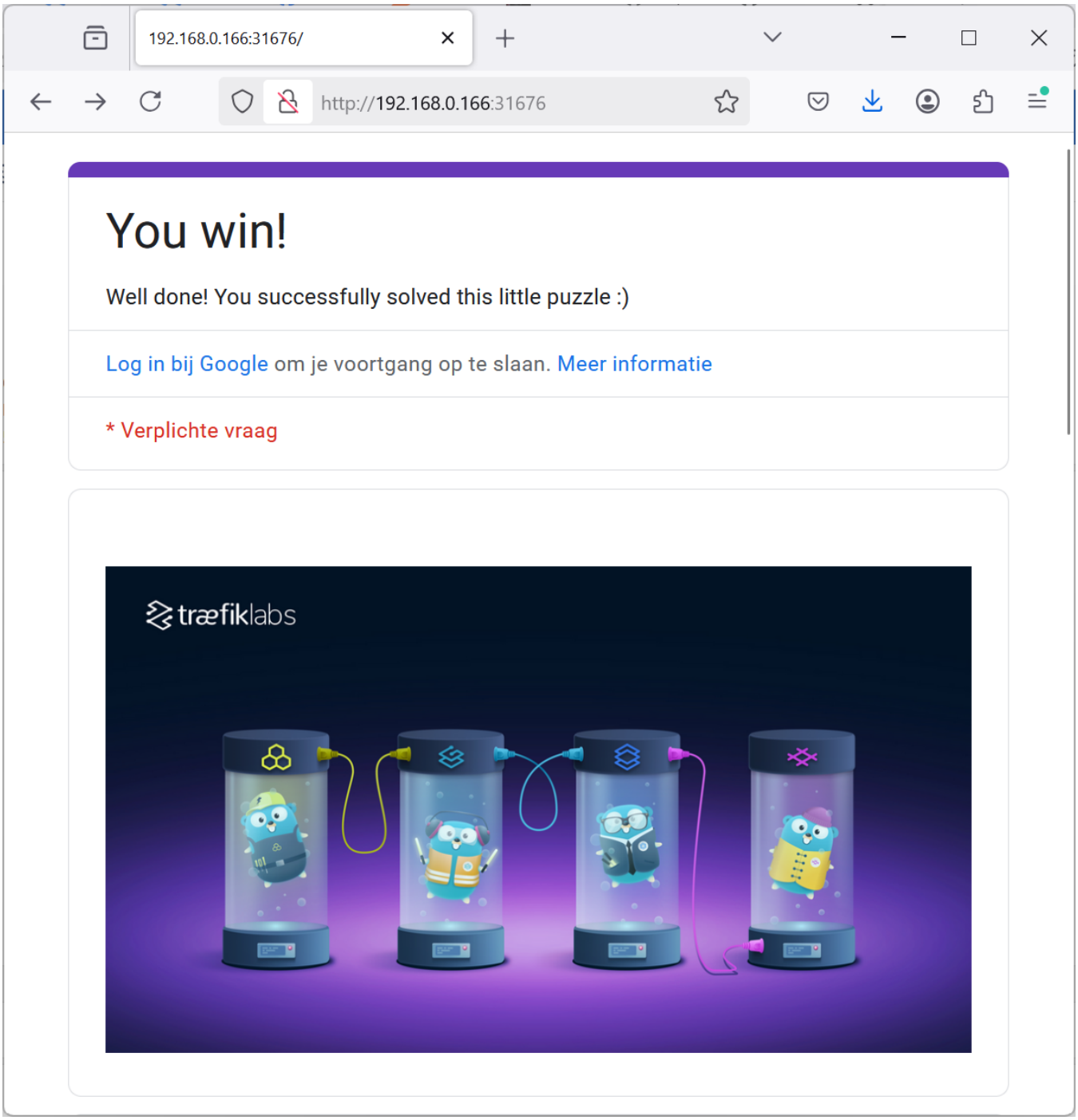
I did already do an inspect of the image, and noticed this:

```
"traefik": "dcc9c530767c102764d45d621fc92317"
```

Let's create a secret.

```
$ kubectl create secret generic traefik --from-literal=traefik=dcc9c530767c102764d45d621fc92317
```

Then simply restart the pod, and check again:


192.168.0.166:31676/
http://192.168.0.166:31676
You win!
Well done! You successfully solved this little puzzle :)
Log in bij Google om je voortgang op te slaan. Meer informatie
* Verplichte vraag
træfiklabs

Hooray!