

A (quick) introduction to TraefikEE





<https://jlevesy.github.io/slides/docker-meetup-lyon-may-2019>

How to use these slides?

- **Browse the slides:** Use the arrows
 - Change chapter: Left/Right arrows
 - Next or previous slide: Top and bottom arrows
- **Overview of the slides:** keyboard's shortcut "o"
- **Speaker mode (and notes):** keyboard's shortcut "s"

whoami

- Julien Levesy
 - TraefikEE developer @containous (we're hiring!)
 -  @jlevesy
 -  jlevesy

What is Traefik ?

The world of production is changing

- Containers as a standard execution unit
- Era of orchestrators and flexible workloads
- Everything goes dynamic

What about reverse proxies ?

- Static configuration in a dynamic environment
- HTTPS is possible, yet painful
- Lack of observability

Say hello to Traefik

- Cheese based HTTP/HTTPS reverse proxy
- Dynamic at its core
- Versatile
- Highly observable
- Let's encrypt built-in
- Advanced features out of the box

All included into a single binary

Open Source & Written in Golang

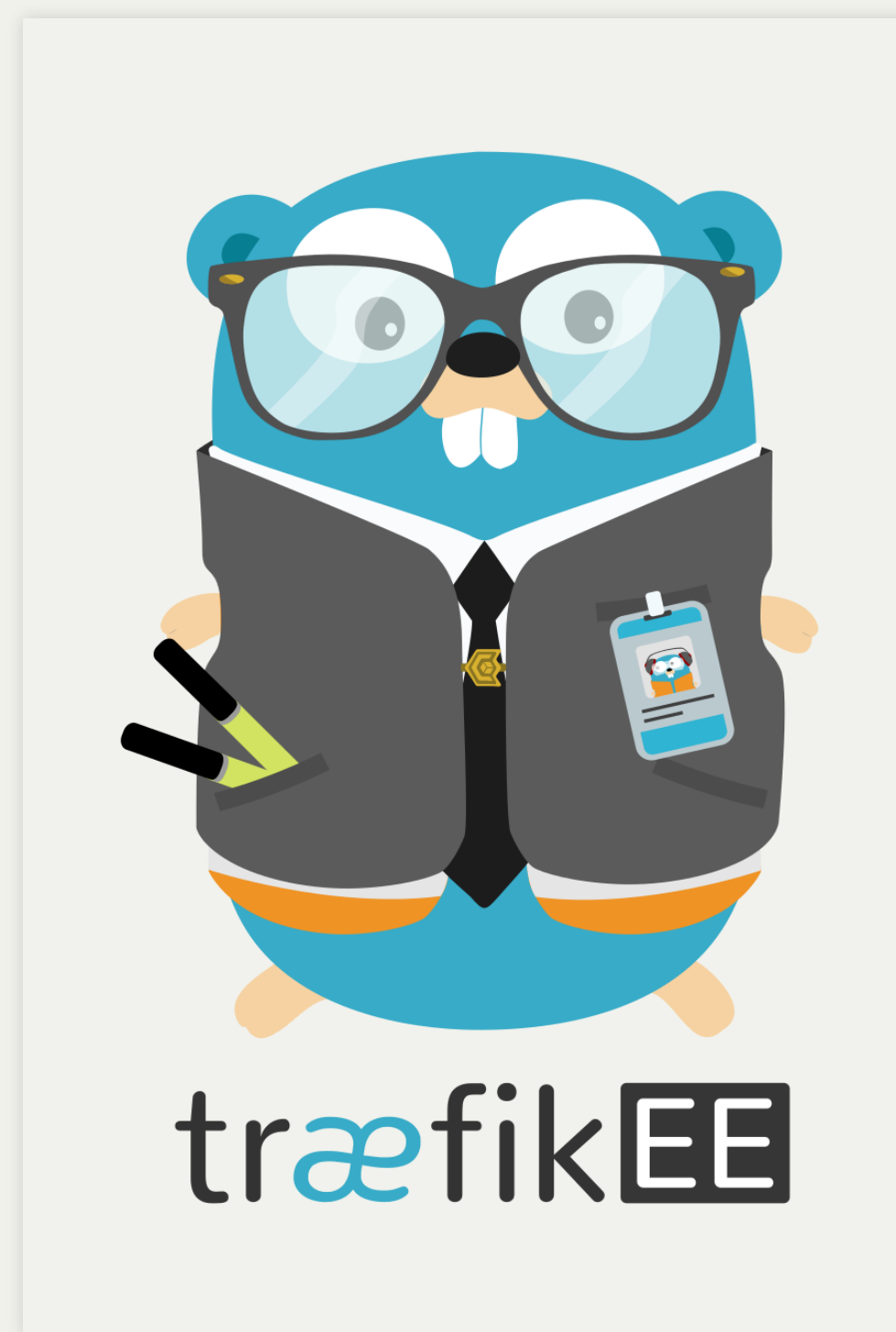
Contributions are welcome !

And it gets better with time !

- v2.0 in alpha stage
- Flexible architecture
- TCP support
- Updated Kubernetes ingress configuration based on CRD
- Improved routing rule syntax
- And more to come !

How To Make Traefik Ready For The Enterprise ?

We Gave It a Suit !



What a Company Wants From a Reverse Proxy ?

- Reliability
- Security
- Full support
- All the features from Traefik !
- Scalability

So we Made Traefik Distributed !

FROM THE INTERNET

TO YOUR INFRASTRUCTURE



Control Plane



Data Plane



And There Is More !

- No restart needed to configure the proxy
- Backup and restore system
- Distributed features out of the box
 - Let's encrypt support
 - Rate limiting (1.1)
 - Max connections (1.1)
 - Circuit breaker (1.2)

All Included Into a Single Binary

(No need of an external key value store)

Actually, There Is Another Binary

`traefikeectl` our CLI

- Installs TraefikEE automatically on k8s and swarm
- Deploys configuration
- Get cluster informations
- Create backups
- Fetch the logs

Docker Swarm Demo

Our Infrastructure

- Docker Swarm cluster
 - 3 managers
 - 3 workers
- Created using `sind`

Install TraefikEE

```
traefikeectl install --swarm --dashboard --licensekey=${TRAEFIKEE_LICENSE_KEY}
```

Deploy an App

```
version: '3.7'
networks:
  traefikee_net:
    name: traefikee_net
    external: true

services:
  whoami:
    image: jlevesy/webapp:latest
    deploy:
      mode: replicated
      replicas: 2
      labels:
        - "traefik.frontend.entryPoints=http,https"
        - "traefik.frontend.rule=Host:app.traefikee.io"
        - "traefik.port=80"
    networks:
      - traefikee_net
```

Operate TraefikEE

```
# enable TLS on 443
traefikeectl deploy \
  --docker.swarmmode \
  --entryPoints='Name:http Address::80' \
  --entryPoints='Name:https Address::443 TLS'

# enable the metrics on the http port
# (even if that is something you don't want to do !)
traefikeectl deploy --docker.swarmmode --metrics.prometheus.entrypoint=http
```

And When You're Done

```
# Uninstall traefikee  
traefikeectl uninstall  
  
# Delete the swarm cluster  
sind delete  
  
# Reconfigure your docker client  
unset DOCKER_HOST
```


Thanks

🐦 @jlevesy 🐙 jlevesy



- Slides (HTML): <https://jlevesy.github.io/slides/docker-meetup-lyon-may-2019>
- Slides (PDF): <https://jlevesy.github.io/slides/docker-meetup-lyon-may-2019/slides.pdf>
- Source on 🐙: <https://github.com/jlevesy/slides/tree/docker-meetup-lyon-may-2019>