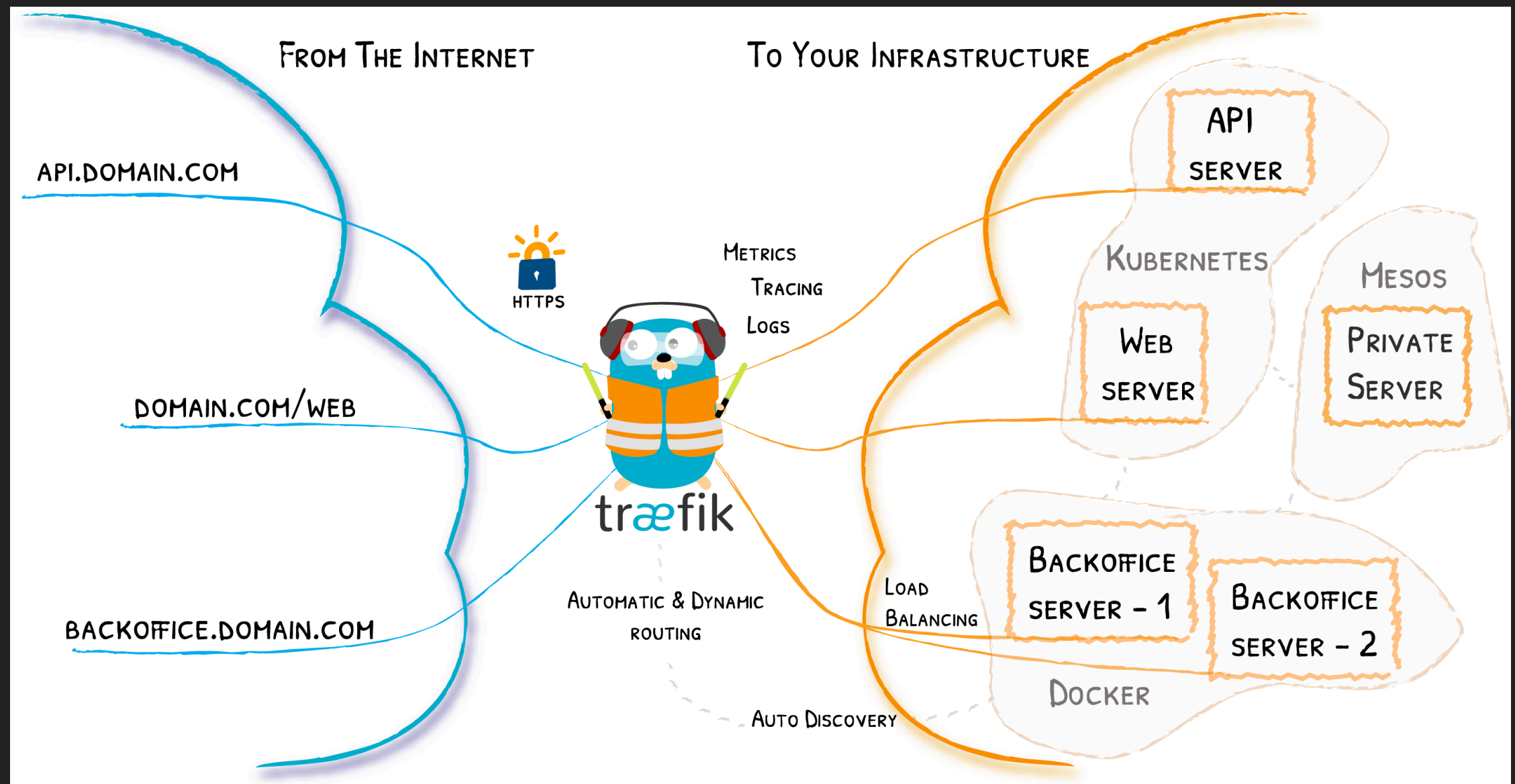


ДЛЯ ФРОНТЕНДЕРОВ

ПРОЩЕ, ЧЕМ NGINX

КАК ЭТО ВСЕ ВЫГЛЯДИТ



О ЧЁМ ПОЙДЁТ РЕЧЬ?

- ▶ Зачем мне traefik?
- ▶ Пару слов про настройку
- ▶ Настраиваем traefik.toml
- ▶ Настраиваем docker-compose
- ▶ Как с этим работать
- ▶ TL;DR

**А ЗАЧЕМ МНЕ
TRAFFIC?**

Nginx

```

...

http {
    include conf/mime.types;
    include /etc/nginx/proxy.conf;
    include /etc/nginx/fastcgi.conf;
    index index.html index.htm index.php;
    ...

    server { # php/fastcgi
        listen 80;
        server_name domain1.com www.domain1.com;
        access_log logs/domain1.access.log main;
        root html;

        location ~ \.php$ {
            fastcgi_pass 127.0.0.1:1025;
        }
    }

    server { # simple reverse-proxy
        listen 80;
        server_name domain2.com www.domain2.com;
        access_log logs/domain2.access.log main;

        # serve static files
        location ~ ^/(images|javascript|js|css|flash|media|static)/ {
            root /var/www/virtual/big.server.com/htdocs;
            expires 30d;
        }

        # pass requests for dynamic content to rails/turbogears/zope, et al
        location / {
            proxy_pass http://127.0.0.1:8080;
        }
    }

    ...

```

VS

traefik

```

defaultEntryPoints = ["http", "https"]

[entryPoints]
[entryPoints.http]
    address = ":80"
    [entryPoints.http.redirect]
        entryPoint = "https"
[entryPoints.https]
    address = ":443"
    [entryPoints.https.tls]

[acme]
email = "test@your_domain"
storage = "acme.json"
onHostRule = true
entryPoint = "https"
    [acme.httpChallenge]
        entryPoint = "http"

[docker]
domain = "YOUR_DOMAIN"
watch = true
network = "web"

exposedbydefault = false

```

ПЛЮСЫ TRAEFIK

- ▶ один раз настроили и забыли:
автоматический роутинг ваших приложений
в реальном времени без постоянного
редактирования конфигурации;
- ▶ интеграция с Let's Encrypt (https), Docker,
Kubernetes, e.t.c...;
- ▶ подробная документация

NGINX VS TRAEFIK? МИНУСЫ TRAEFIK

- ▶ оверкилл для одного статичного сайта;
- ▶ сообщество меньше;
- ▶ бенчмарки: траффик медленнее nginx. Но ненамного (15%);

ПАРА СЛОВ ПРО НАСТРОЙКУ

НАМ ПОНАДОБЯТСЯ:

- ▶ Docker

[Пример установки докера](#)

- ▶ docker-compose

[Пример установки docker-compose](#)

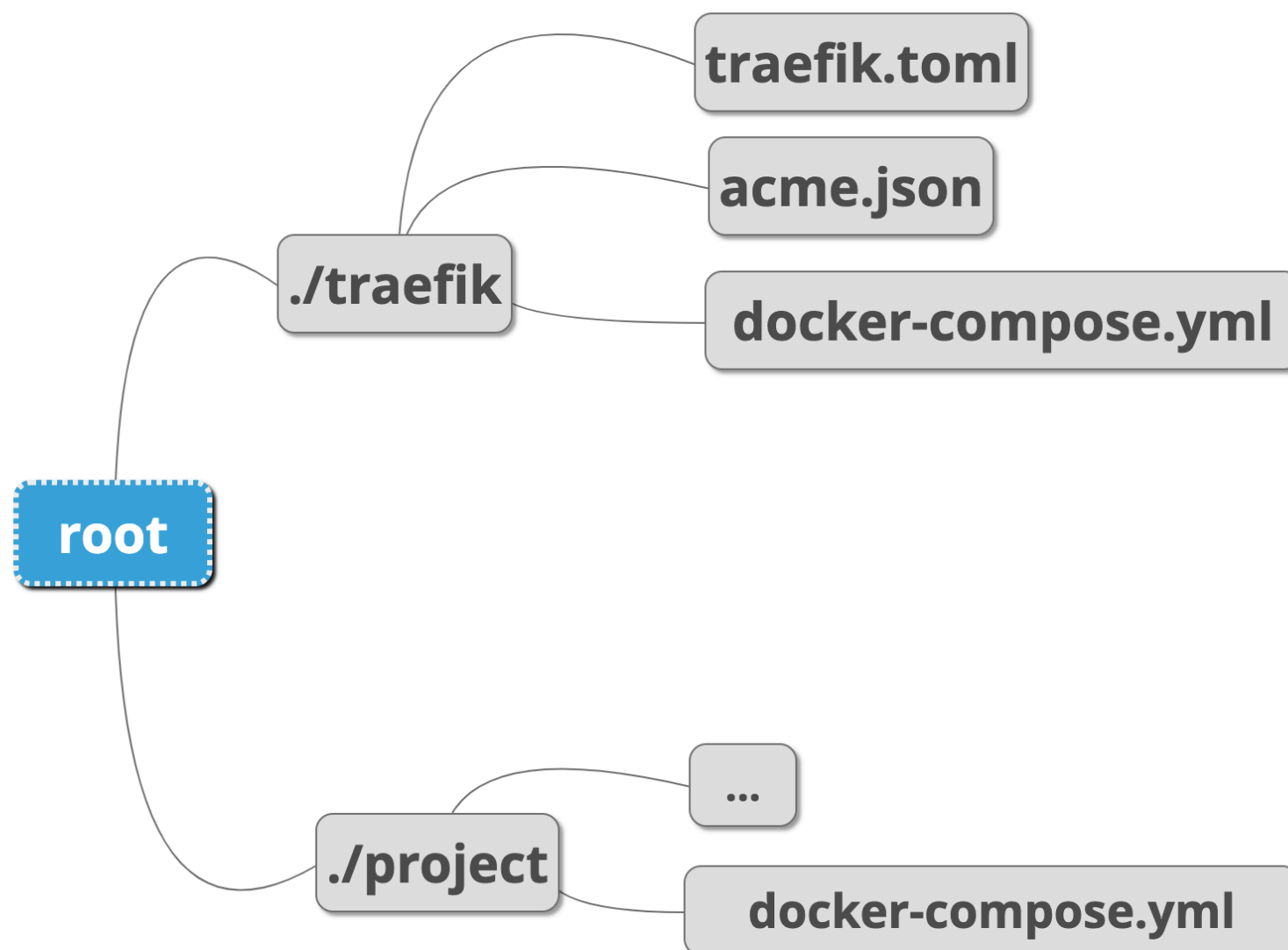
- ▶ А-записи домена и поддомена

Домен далее я обозначу как YOUR_DOMAIN

- ▶ traefik

[Пример настройки traefik](#)

ПРИМЕР СТРУКТУРЫ



НАСТРАИВАЕМ
TRAFFIK.TOML

КАК ВЫГЛЯДИТ КОНФИГ?

Определим базовую конфигурацию для Traefik.

[Документация.](#)

Это минимальные настройки.

Мы задаём правила для редиректа http на https, настраиваем Let's Encrypt и интегрируемся с докером.

```
defaultEntryPoints = ["http", "https"]

[entryPoints]
  [entryPoints.http]
    address = ":80"
    [entryPoints.http.redirect]
      entryPoint = "https"
  [entryPoints.https]
    address = ":443"
    [entryPoints.https.tls]

[acme]
email = "test@your_domain"
storage = "acme.json"
onHostRule = true
entryPoint = "https"
  [acme.httpChallenge]
    entryPoint = "http"

[docker]
domain = "YOUR_DOMAIN"
watch = true
network = "web"

exposedbydefault = false
```

LET'S ENCRYPT

- ▶ `email`: указываем действующую почту, которая будет использоваться для генерации сертификатов
- ▶ `onHostRule: true` – сертификаты будут сгенерированы сразу после создания контейнера с заданным именем хоста
- ▶ Что такое `httpChallenge`

Про Let's Encrypt и traefik

```
[acme]
email = "test@your_domain"
storage = "acme.json"
onHostRule = true
entryPoint = "https"
  [acme.httpChallenge]
    entryPoint = "http"
```

ИНТЕГРАЦИЯ С DOCKER

- ▶ `network`: traefik будет следить за контейнерами в одной сети, допустим, назовём ее `web`.
- ▶ `exposedbydefault: false` – traefik будет реагировать на контейнеры, которые явно помечены с помощью лейблов. Об этом дальше

Как настроить `network` для traefik?

```
[docker]
domain = "YOUR_DOMAIN"
watch = true
network = "web"

exposedbydefault = false
```

НАСТРАИВАЕМ DOCKER-COMPOSE

КАК ВЫГЛЯДИТ КОНФИГ?

Чтобы **traefik** подписался на создание контейнеров, сделаем ему собственный контейнер и включим его в сеть **web**.

[Подробная настройка](#)

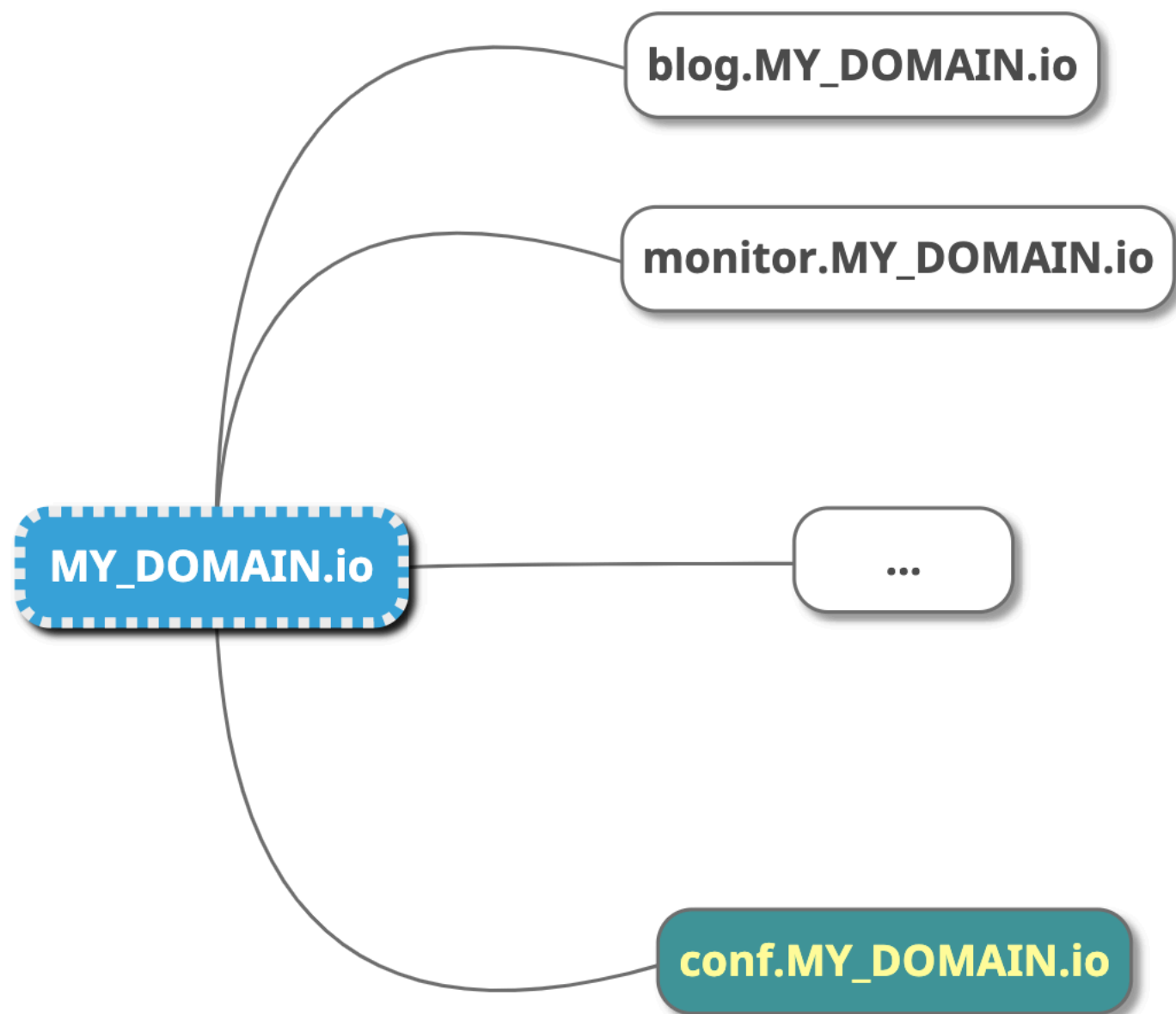
```
version: '2'

services:
  traefik:
    image: traefik
    restart: always
    ports:
      - 80:80
      - 443:443
    networks:
      - web
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock
      - /opt/traefik/traefik.toml:/traefik.toml
      - /opt/traefik/acme.json:/acme.json
    container_name: traefik

networks:
  web:
    external: true
```


**КАК С ЭТИМ
РАБОТАТЬ**

ЧТО Я ХОЧУ



ВОТ МОЙ КОНФИГ

Я добавила новый
контейнер в конфигурацию

[Посмотреть полный конфиг](#)

docker-compose.yml для проекта

```
version: '3'

services:
  ...
  nginx:
    image: nginx:latest
    container_name: nginx
    networks:
      - web
    volumes:
      - ./site:/usr/share/nginx/html
    labels:
      - traefik.enable=true
      - traefik.frontend.rule=Host:yepstepz.io
      - traefik.port=80
  conf:
    image: nginx:latest
    container_name: conf
    networks:
      - web
    volumes:
      - ./conf:/usr/share/nginx/html
    labels:
      - traefik.enable=true
      - traefik.frontend.rule=Host:conf.yepstepz.io
      - traefik.port=80
networks:
  web:
    external: true
```

ПОСЛЕДНИЕ ШТРИХИ

```
docker-compose up -d
```

**TL;DR: TAK ЗАЧЕМ
BAM TRAEFIK**

КРАТКО:

- ▶ Девиз Traefik: "Настрой один раз и забудь". Не нужно перезагружать сервер, редактировать конфиг. Приложения поднимаются на урлах "на лету".
- ▶ Traefik легко интегрируется с Let's Encrypt: не нужно генерировать для доменов и поддоменов дополнительные **https** сертификаты.
- ▶ Traefik можно использовать без Docker и Kubernetes. Но с ними жизнь легче, их всегда можно подключить.



**СПАСИБО ЗА
ВНИМАНИЕ!**