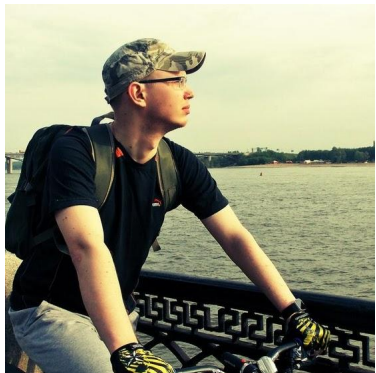


# Docker



Использование Docker в pet-проекте

# Даниил Кузнецов



- 5+ лет опыта backend разработки
- highload
- Docker
- Go

# Цель: разобраться с Docker

Блог:

- легкий перенос данных между хостингами
- ssl
- Docker 17.06
- Docker swarm +
- Docker stack ->
- docker-compose.yml v3

# Hugo



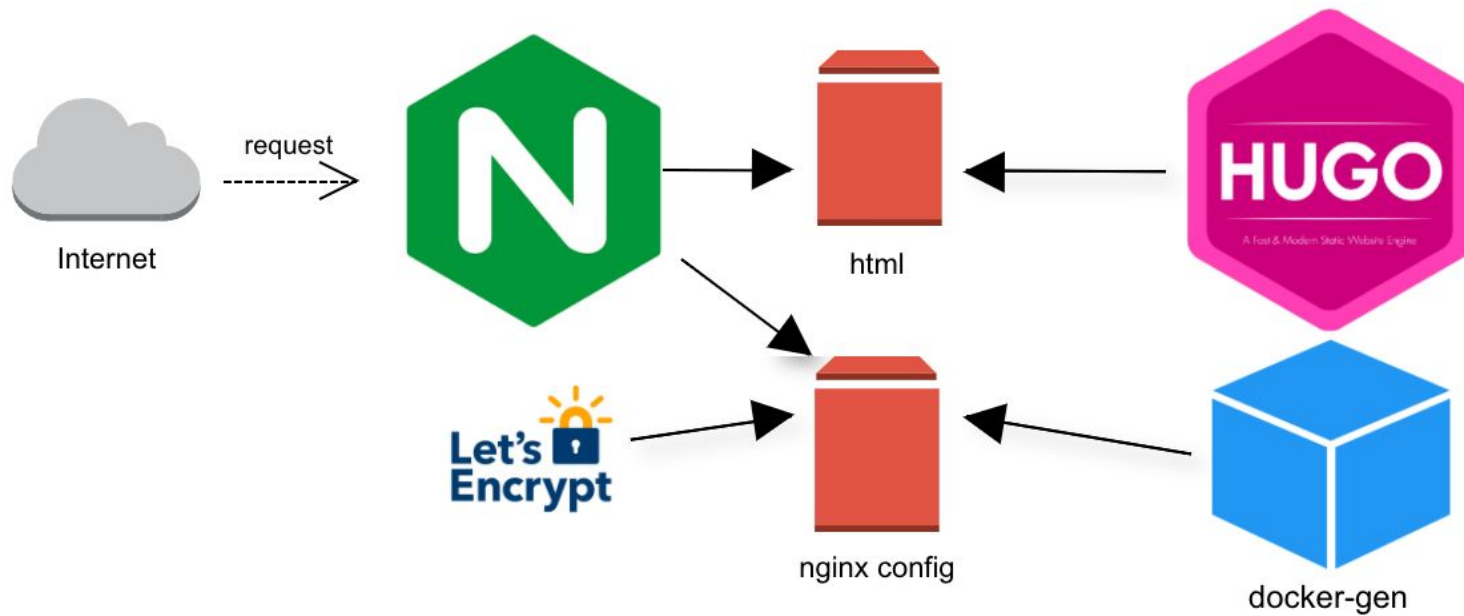
- генератор статичных сайтов
- написан на Go
- обширная документация
- 100+ тем оформления
- исходники статей в markdown разметке
- простая интеграция ga, discuss, etc

<https://gohugo.io/>

# Nginx, docker-gen, Let's encrypt

- docker-gen
  - написан на Go
  - реагирование на события docker
  - генерация конфигов по шаблонам и env переменным
- letsencrypt companion
  - получение SSL-сертификатов
- nginx
  - раздача контента

# Схема I



# Решение I

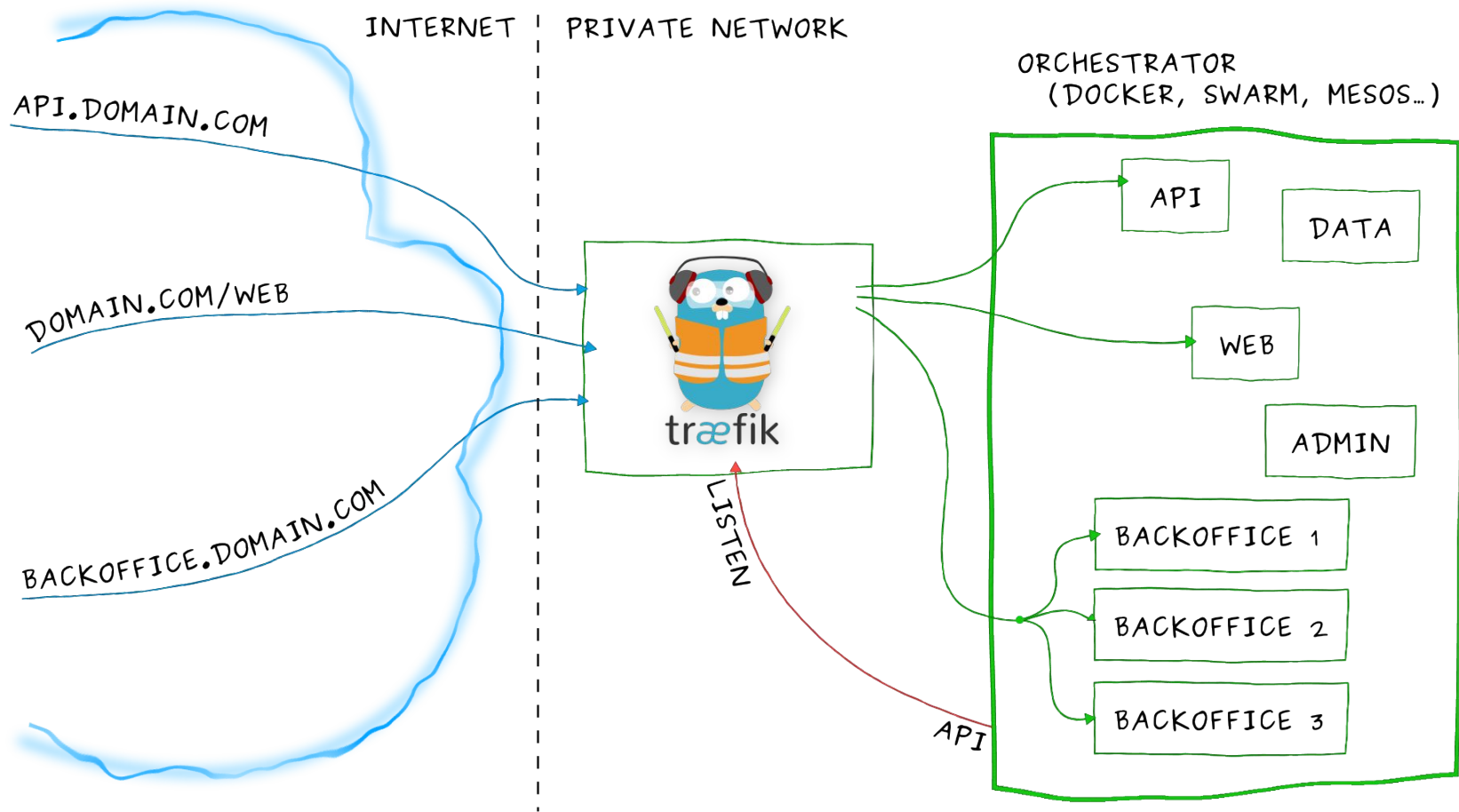
- + docker-compose up - все работает
- + ssl
- + динамические конфиги nginx на основе переменных
- отсутствие поддержки docker stack

## Решение II

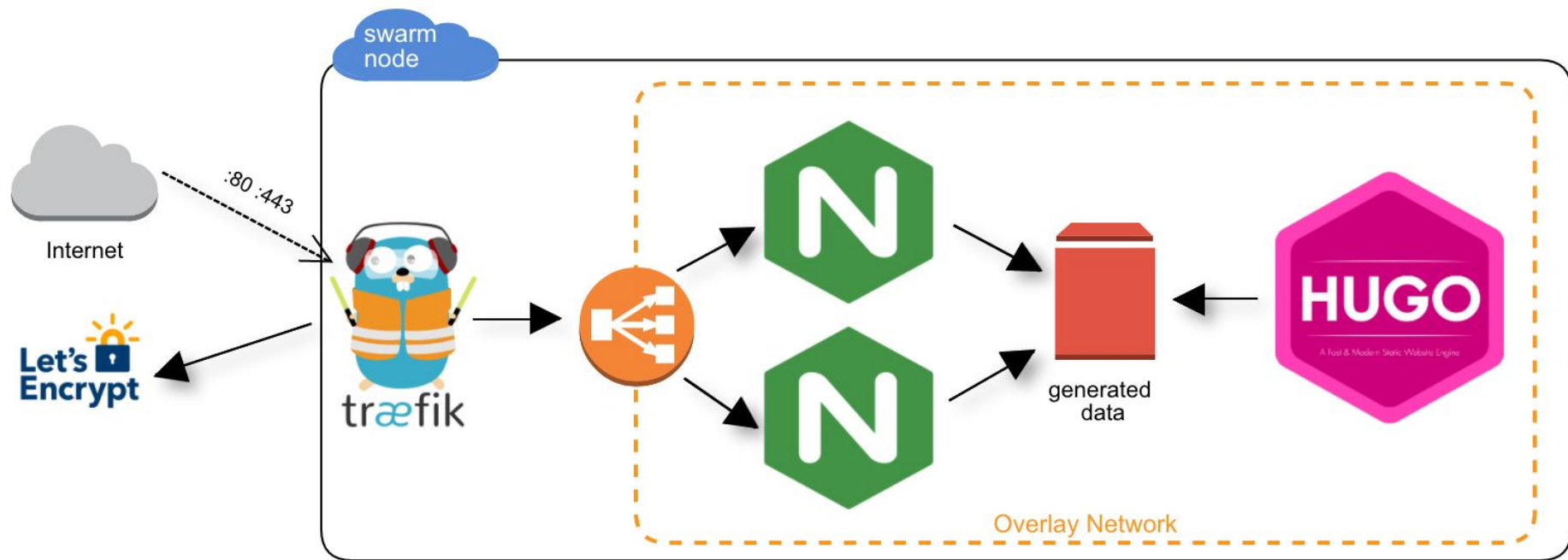


- HTTP reverse proxy
- написан на Go :)
- поддержка Docker, Docker swarm mode, k8s и тд.
- поддержка ssl от Let's encrypt
- роутинг запросов по правилам
- простота настройки и gui





## Схема II



## Решение II

- + ssl
- + Docker stack + Docker swarm
- + обширная документация по traefik
- + Docker swarm load balancer

# Итоговое решение

- сервисы
  - nginx (official image)
  - traefik (official image)
  - hugo (custom image)
- docker
  - stack
  - compose v3
  - overlay network
- статьи в .md

# Workflow

- пишем текст в .md
- make docker-build
- push образа в docker registry
- ssh root@somehostname
- обновляем сервис с hugo
  - **docker service update --force --detach=false hugo-service-name**

# Автоматизируй это!

1. выполнить webhook при пуше в репозиторий (**github.com**)
2. webhook handler
3. `hugo -s dir_with_md_files -o nginx_volume`

<https://github.com/adnanh/webhook>

- lightweight configurable tool written in Go, that allows you to easily create HTTP endpoints (hooks) on your server

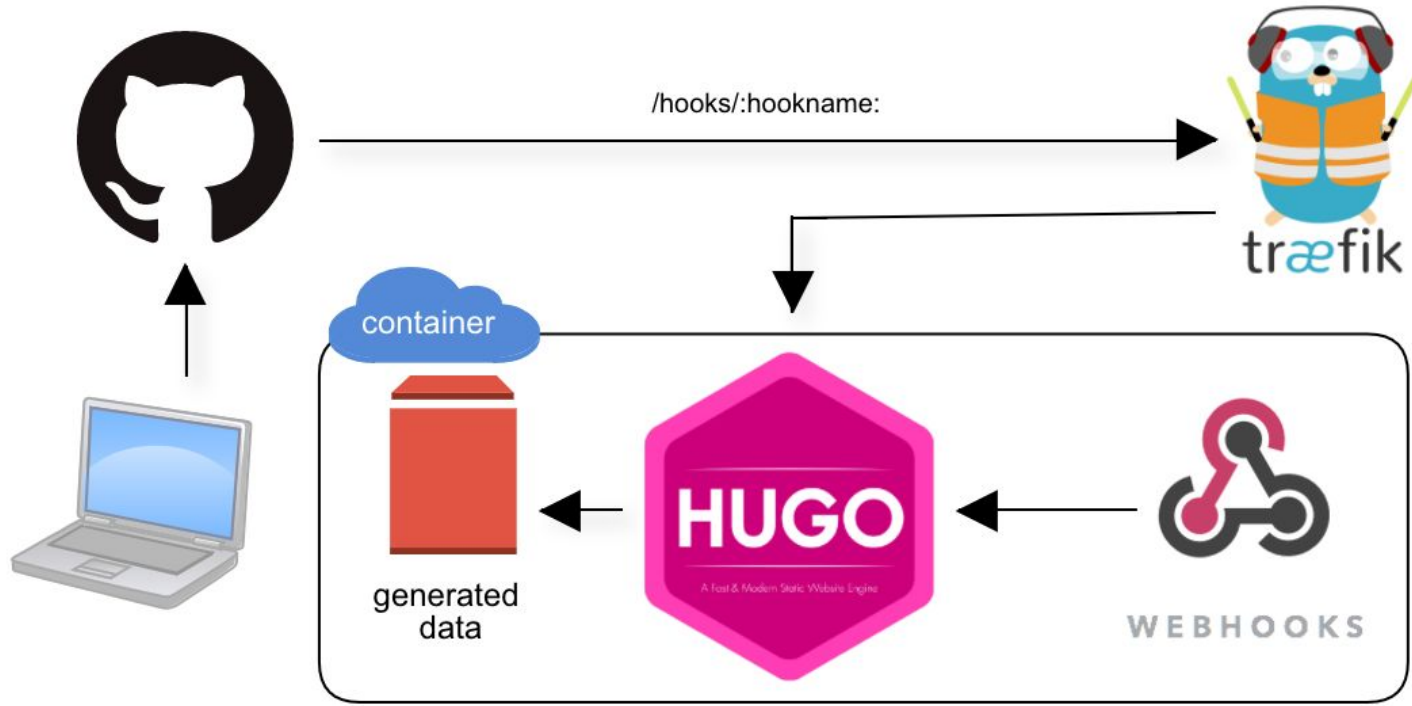
# Hooks.json

<https://d2one.ru/hooks/redeploy>

```
[
  {
    "id": "redeploy",
    "execute-command": "/run.sh",
    "pass-arguments-to-command": [
      { "source": "string", "name": "webhook" }
    ]
  }
]
```

/run.sh webhook

# Webhook + traefik + github





# Hugo + Webhook image. Multi-stage

```
FROM golang:1.8-alpine AS builder  
RUN apk add --no-cache git  
RUN go get -v github.com/adnanh/webhook  
RUN go get -v github.com/spf13/hugo
```

.....

# Hugo + Webhook image

.....

FROM alpine:3.5

RUN apk add --no-cache bash git && rm -rf /var/cache/apk/\*

COPY --from=builder /go/bin/ /usr/local/bin/

.....

# Hugo + Webhook image

.....  
COPY hooks.json /etc/webhook/hooks.json  
COPY run.sh /run.sh

EXPOSE 9000  
ENTRYPOINT ["/bin/bash", "/run.sh"]  
CMD ["run"]

# Traefik service

```
command: -c /etc/traefik/traefik.toml --web --docker --docker.swarmmode  
        --docker.watch --docker.domain=d2one.ru
```

```
networks:
```

- traefik-net

```
ports:
```

- "80:80"
- "443:443"

```
volumes:
```

- /var/run/docker.sock:/var/run/docker.sock
- \$PWD/volumes/traefik/:/etc/traefik/

```
deploy:
```

```
  placement:
```

```
    constraints: [node.role == manager]
```

# Nginx service

networks:

- traefik-net

deploy:

mode: replicated

replicas: 3

labels:

- "traefik.port=80"
- "traefik.docker.network=traefik-net"
- "traefik.backend=nginx"
- "traefik.frontend.rule=Host:d2one.ru"
- "traefik.backend.loadbalancer.method=drp"
- "traefik.backend.loadbalancer.swarm=true"

# Hugo service

environment:

- HUGO\_REPO\_URL=https://github.com/d2one/d2one.github.io.git
- NGINX\_HTML\_VOLUME=/usr/share/nginx/html/

deploy:

mode: replicated

replicas: 1

labels:

- "traefik.port=9000"
- "traefik.docker.network=traefik-net"
- "traefik.backend=hugo"
- "traefik.frontend.rule=PathPrefix:/hooks/"
- "traefik.backend.loadbalancer.method=drp"
- "traefik.backend.loadbalancer.swarm=true"

# Traefik webpanel

## frontend-Host-d2one-ru

Route	Rule
route-frontend-Host-d2one-ru	Host:d2one.ru

[http](#) [https](#) [Backend:backend-nginx](#) [PassHostHeader](#)

## frontend-PathPrefix-hooks

Route	Rule
route-frontend-PathPrefix-hooks	PathPrefix:/hooks/

[http](#) [https](#) [Backend:backend-hugo](#) [PassHostHeader](#)

## backend-hugo

Server	URL	Weight
server-traefic-reverse_hugo	<a href="http://10.0.0.2:9000">http://10.0.0.2:9000</a>	0

[Load Balancer: drr](#)

## backend-nginx

Server	URL	Weight
server-traefic-reverse_nginx	<a href="http://10.0.0.4:80">http://10.0.0.4:80</a>	0

[Load Balancer: drr](#)



Daniil Kuznetsov

Programmer

## Инструменты web-разработчика

## Интересные ссылки

## Чем занимаются backend-разработчики?

## MongoDB. Часть 5. Индексы



```
└─$ hey -n 10000 -c 100 https://d2one.ru
```

Total: 22.3696 secs

Slowest: 5.4829 secs

Fastest: 0.0890 secs

Average: 0.2103 secs

Requests/sec: 447.0361

Total data: 56010000 bytes

Size/request: 5601 bytes

[illegible]

# ИСТОЧНИКИ

- <https://gohugo.io/>
- <https://github.com/adnanh/webhook>
- <https://d2one.ru/>
- <https://github.com/d2one/docker-training>
- <https://nginx.ru/ru/>
- <https://traefik.io/>

# Вопросы?

