PHP on Docker

윤영진
yupmin@gmail.com

방법 1

docker NGINX

> deploy

docker phpfpm

PECL
The PHP Extension Library

volume mount

PHP Code

```php
<?php
require 'PHPMailerAutoload.php';

$mail = new PHPMailer;

//$mail->SMTPDebug = 3;                               // Enable verbose debug output

$mail->isSMTP();                                      // Set mailer to use SMTP
$mail->Host = 'smtp1.example.com;smtp2.example.com';  // Specify main and backup SMTP servers
$mail->SMTPAuth = true;                               // Enable SMTP authentication
$mail->Username = 'user@example.com';                 // SMTP username
$mail->Password = 'secret';                           // SMTP password
$mail->SMTPSecure = 'tls';                            // Enable TLS encryption, `ssl` also accepted
$mail->Port = 587;                                    // TCP port to connect to

```

방법 2

PHP Code

docker

NGINX

docker

phpfpm

PECL
The PHP Extension Library

```php
<?php
require 'PHPMailerAutoload.php';

$mail = new PHPMailer;

//$mail->SMTPDebug = 3;                               // Enable verbose debug
output

$mail->isSMTP();                                      // Set mailer to use SMTP
$mail->Host = 'smtp1.example.com;smtp2.example.com';  // Specify main and
backup SMTP servers
$mail->SMTPAuth = true;                               // Enable SMTP
authentication
$mail->Username = 'user@example.com';                 // SMTP username
$mail->Password = 'secret';                           // SMTP password
$mail->SMTPSecure = 'tls';                            // Enable TLS
encryption, `ssl` also accepted
$mail->Port = 587;                                    // TCP port to connect to

```

COMPOSER

```
> deploy
```

SEMVER
1.0.0

Docker 꾸겨넣기

- app
- public
- …

- docker : docker assets directory
- Dockerfile
- docker-compose.yml
- .dockerignore

https://github.com/yupmin/laravel-docker-example

Dockerfile

```
FROM php:7.3-fpm-alpine
MAINTAINER yun young jin <yupmin@gmail.com>

RUN apk update && apk add zip git && \
    add --no-cache icu-dev zlib-dev libzip-dev && \
    docker-php-ext-install opcache intl bcmath zip pdo_mysql
RUN cp /usr/local/etc/php/php.ini-production /usr/local/etc/php/php.ini && \
    sed -i "s/display_errors = Off/display_errors = On/" /usr/local/etc/php/php.ini && \
    sed -i "s/upload_max_filesize = .*/upload_max_filesize = 10M/" /usr/local/etc/php/php.ini && \
    sed -i "s/post_max_size = .*/post_max_size = 12M/" /usr/local/etc/php/php.ini && \
    sed -i "s/;cgi.fix_pathinfo=1/cgi.fix_pathinfo=0/" /usr/local/etc/php/php.ini && \
    sed -i "s/variables_order = .*/variables_order = 'EGPCS'/" /usr/local/etc/php/php.ini && \
    sed -i "s/listen = .*/listen = 9000/" /usr/local/etc/php-fpm.d/www.conf && \
    sed -i "s/pm.max_children = .*/pm.max_children = 200/" /usr/local/etc/php-fpm.d/www.conf && \
    sed -i "s/pm.start_servers = .*/pm.start_servers = 56/" /usr/local/etc/php-fpm.d/www.conf && \
    sed -i "s/pm.min_spare_servers = .*/pm.min_spare_servers = 32/" /usr/local/etc/php-fpm.d/www.conf && \
    sed -i "s/pm.max_spare_servers = .*/pm.max_spare_servers = 96/" /usr/local/etc/php-fpm.d/www.conf && \
    sed -i "s/^;clear_env = no$/clear_env = no/" /usr/local/etc/php-fpm.d/www.conf
WORKDIR /var/www/html
COPY . /var/www/html

ENV COMPOSER_ALLOW_SUPERUSER 1
RUN cp .env.example .env && \
    curl -sS https://getcomposer.org/installer | php -- && \
# for more speed
    php composer.phar config -g repos.packagist composer https://packagist.kr && \
    php composer.phar global require hirak/prestissimo && \
# composer install
    php composer.phar install --no-dev --no-scripts && \
# change directory permission
    chown -R www-data:www-data /var/www/html/storage /var/www/html/bootstrap/cache/
```

composer.json

```
{
    "name": "laravel/laravel",
    "type": "project",
    "description": "The Laravel Framework.",
    ...
    "license": "MIT",
    "require": {
        "php": "^7.1.3",
        "ext-intl": "*",
        "ext-bcmath": "*",
        "ext-json": "*",
        "fideloper/proxy": "^4.0",
        "laravel/framework": "5.8.*",
        "laravel/tinker": "^1.0"
    },
    "require-dev": {
        "beyondcode/laravel-dump-server": "^1.0",
        "filp/whoops": "^2.0",
        "fzaninotto/faker": "^1.4",
        "mockery/mockery": "^1.0",
        "nunomaduro/collision": "^3.0",
        "phpunit/phpunit": "^7.5"
    },
    ...
}
```

docker-compose.yml

```yaml
version: '2'
services:
  app:
    build:
      context: ./
      dockerfile: Dockerfile
    working_dir: /var/www/html
    volumes:
      - ./:/var/www/html
#       - /var/www/html
#     env_file:
#       - .env
  web:
    build:
      context: ./docker
      dockerfile: web.dockerfile
    working_dir: /var/www/html
    volumes_from:
      - app
    ports:
      - 18085:80
```

docker/vhost.conf

```
server {
    listen       80;
    listen [::]:80;

    charset utf-8;
    client_max_body_size 320M;

    access_log  /var/log/nginx/access.log combined;
    error_log   /var/log/nginx/error.log error;

    index index.php index.html;
    root /var/www/html/public;

    location / {
        try_files $uri /index.php?$args;
    }

    location ~ \.php$ {
        fastcgi_split_path_info ^(.+\.php)(/.+)$;
        fastcgi_pass app:9000;
        fastcgi_index index.php;
        include fastcgi_params;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        fastcgi_param PATH_INFO $fastcgi_path_info;
    }
}
```

> docker-compose build

> docker-compose up

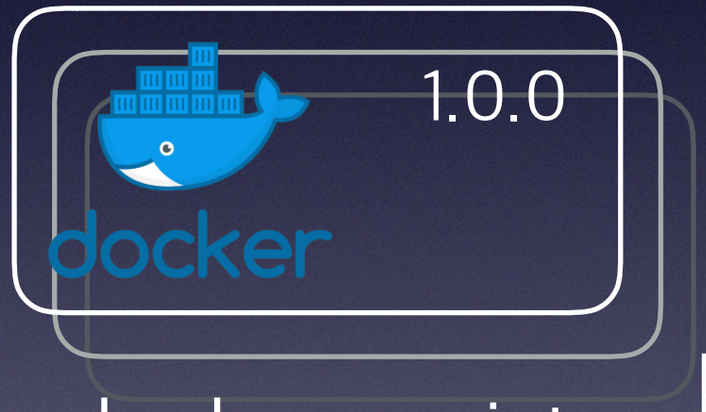
> docker-compose push

to docker repository
with semver

```php
<?php
require 'PHPMailerAutoload.php';

$mail = new PHPMailer;

//$mail->SMTPDebug = 3;                               // Enable verbose debug output

$mail->isSMTP();                                      // Set mailer to use SMTP
$mail->Host = 'smtp1.example.com;smtp2.example.com';  // Specify main and backup SMTP servers
$mail->SMTPAuth = true;                               // Enable SMTP authentication
$mail->Username = 'user@example.com';                 // SMTP username
$mail->Password = 'secret';                           // SMTP password
$mail->SMTPSecure = 'tls';                            // Enable TLS encryption, `ssl` also accepted
$mail->Port = 587;                                    // TCP port to connect to

```

git push

webhook



docker registry

push image

circleci

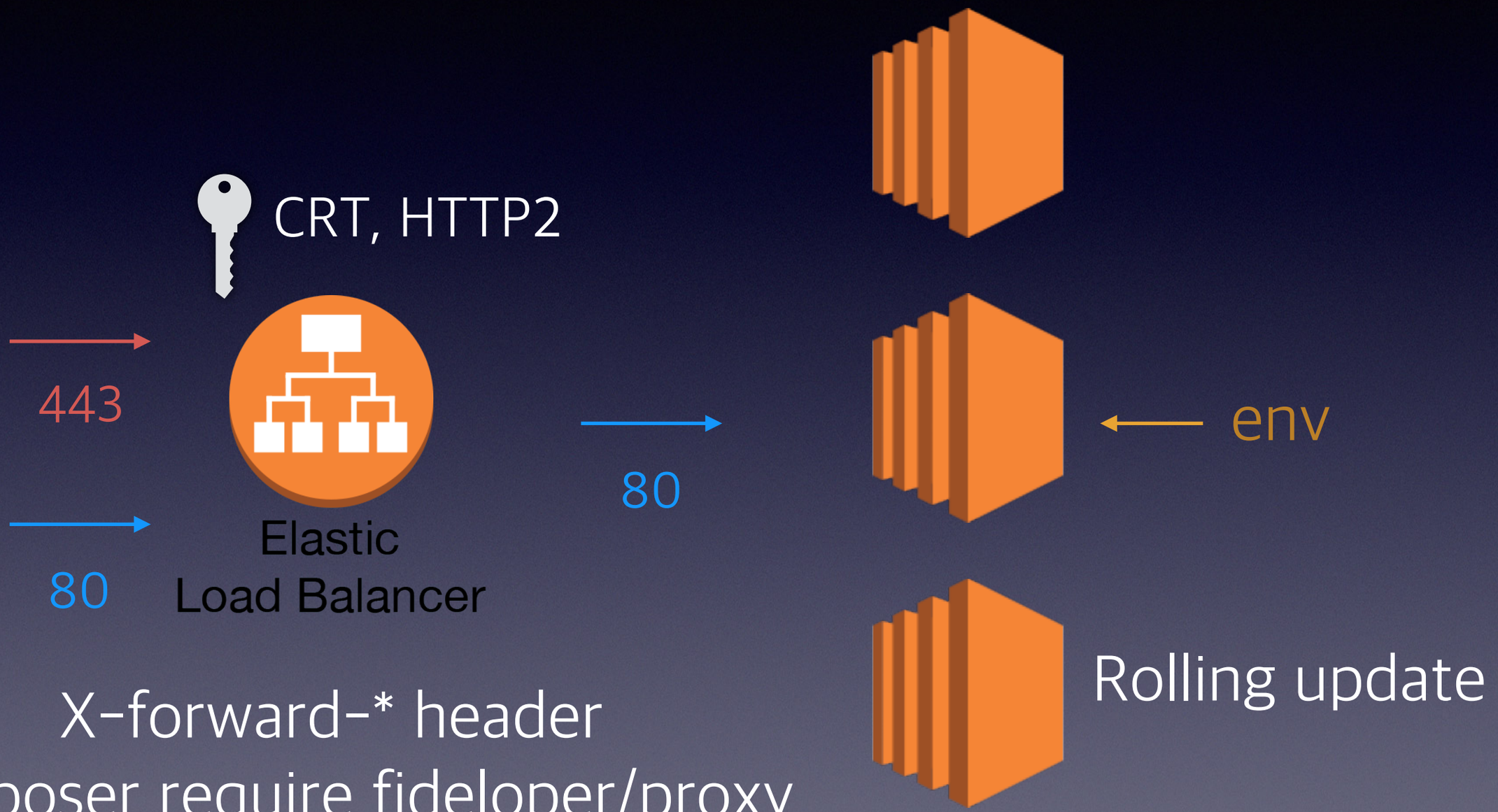
testing & build

circleci

```
> deploy
```

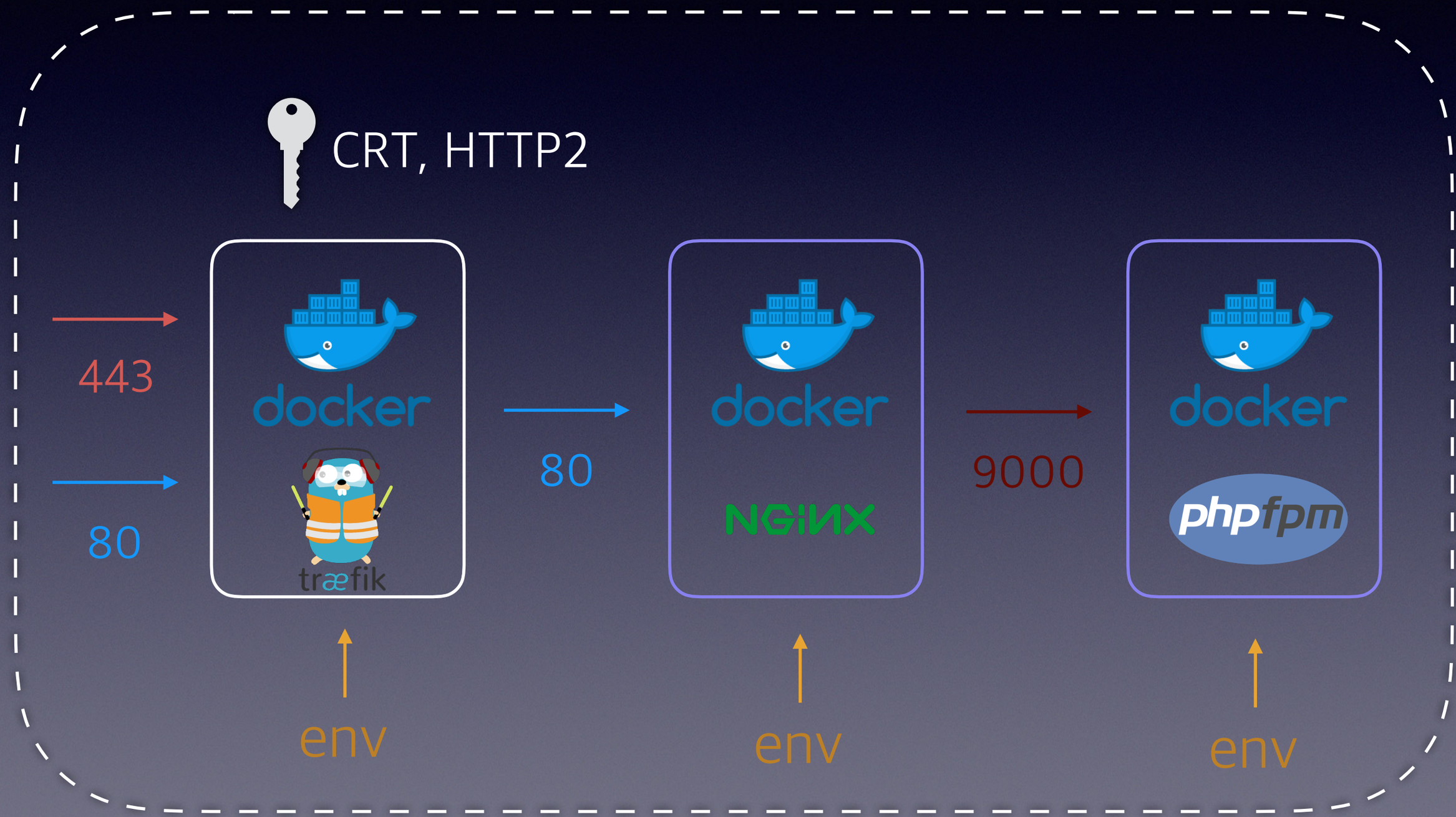
배포간 다운타임!!

AWS Elastic beanstalk

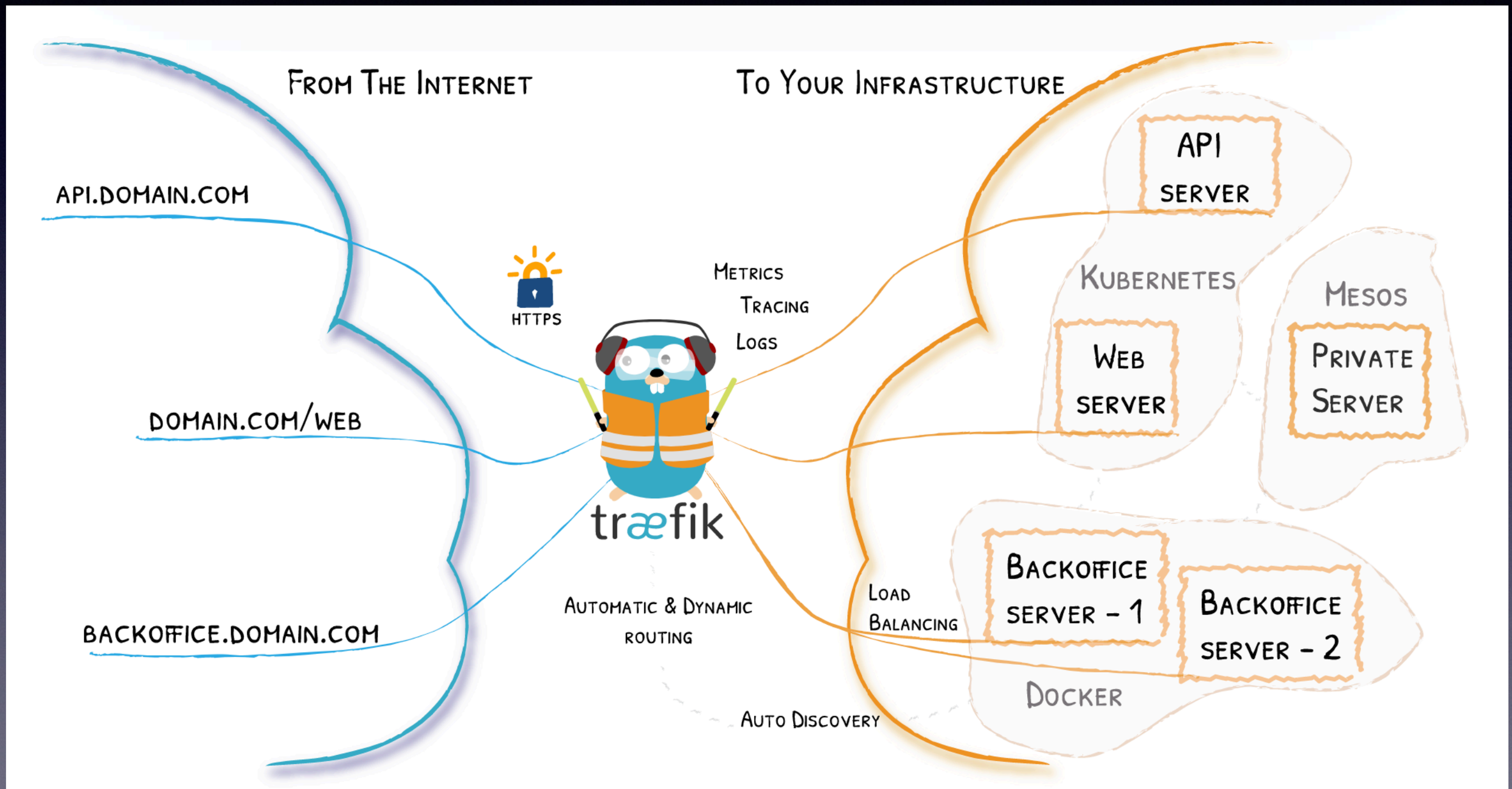


X-forward-* header
composer require fideloper/proxy

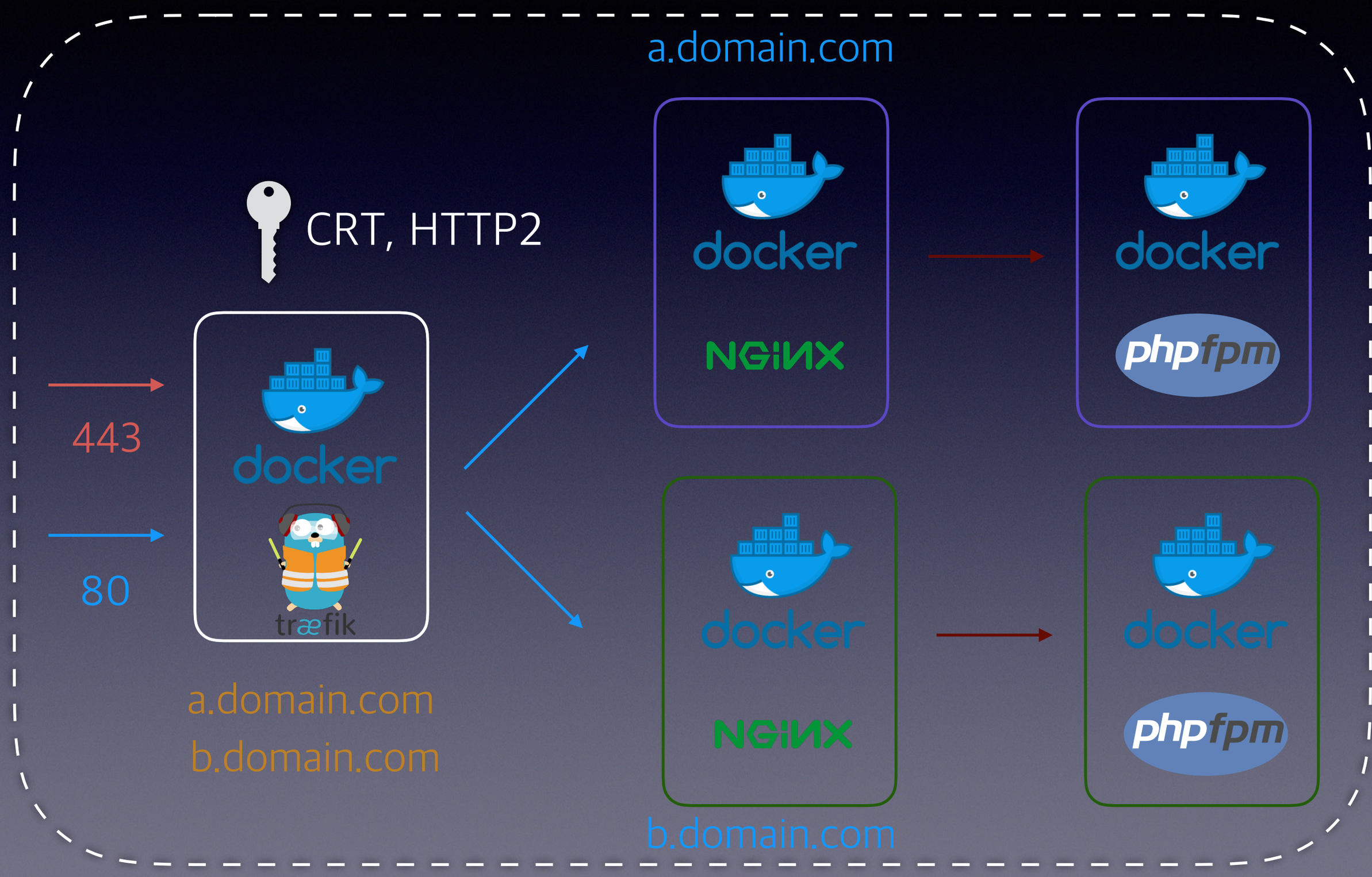
Elastic dockerstalk



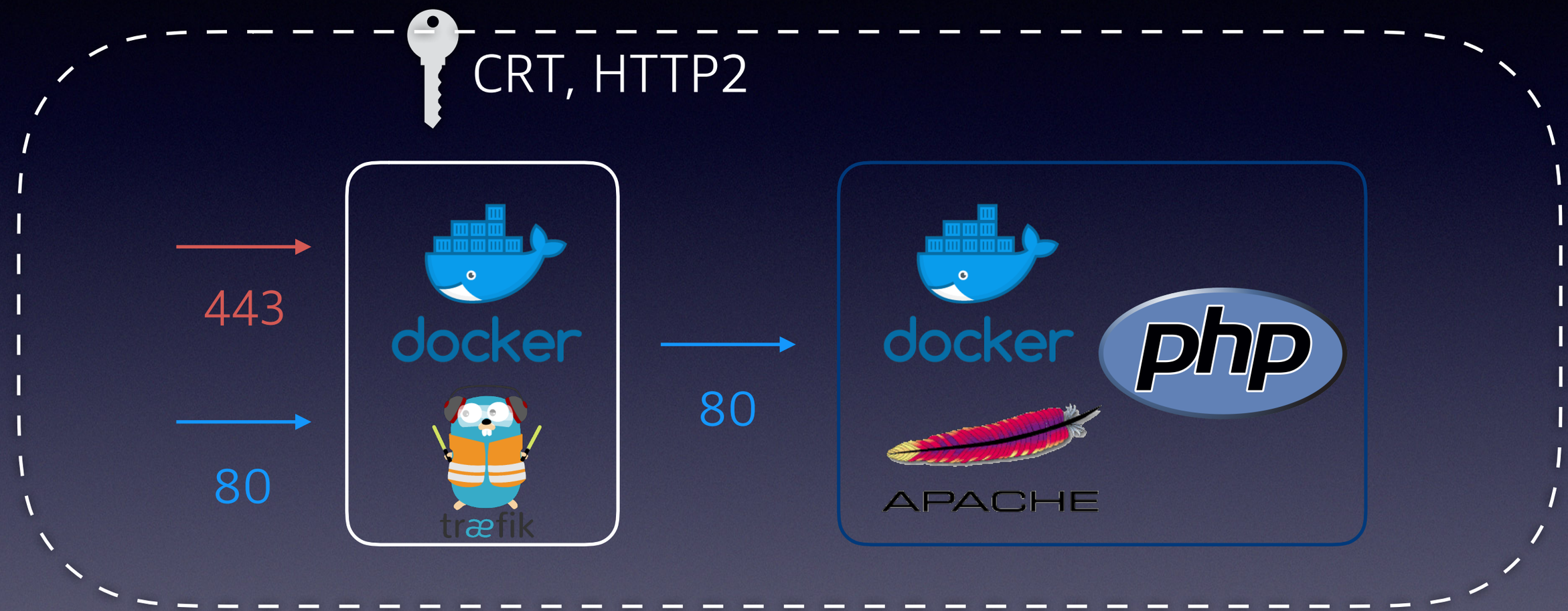
traefik


From The Internet
To Your Infrastructure
API.DOMAIN.COM
DOMAIN.COM/WEB
BACKOFFICE.DOMAIN.COM
HTTPS
Metrics
Tracing
Logs
træfik
Automatic & Dynamic
routing
Load
Balancing
Auto Discovery
API
SERVER
Kubernetes
Web
server
Mesos
Private
Server
Backoffice
server - 1
Backoffice
server - 2
Docker

Elastic dockerstalk



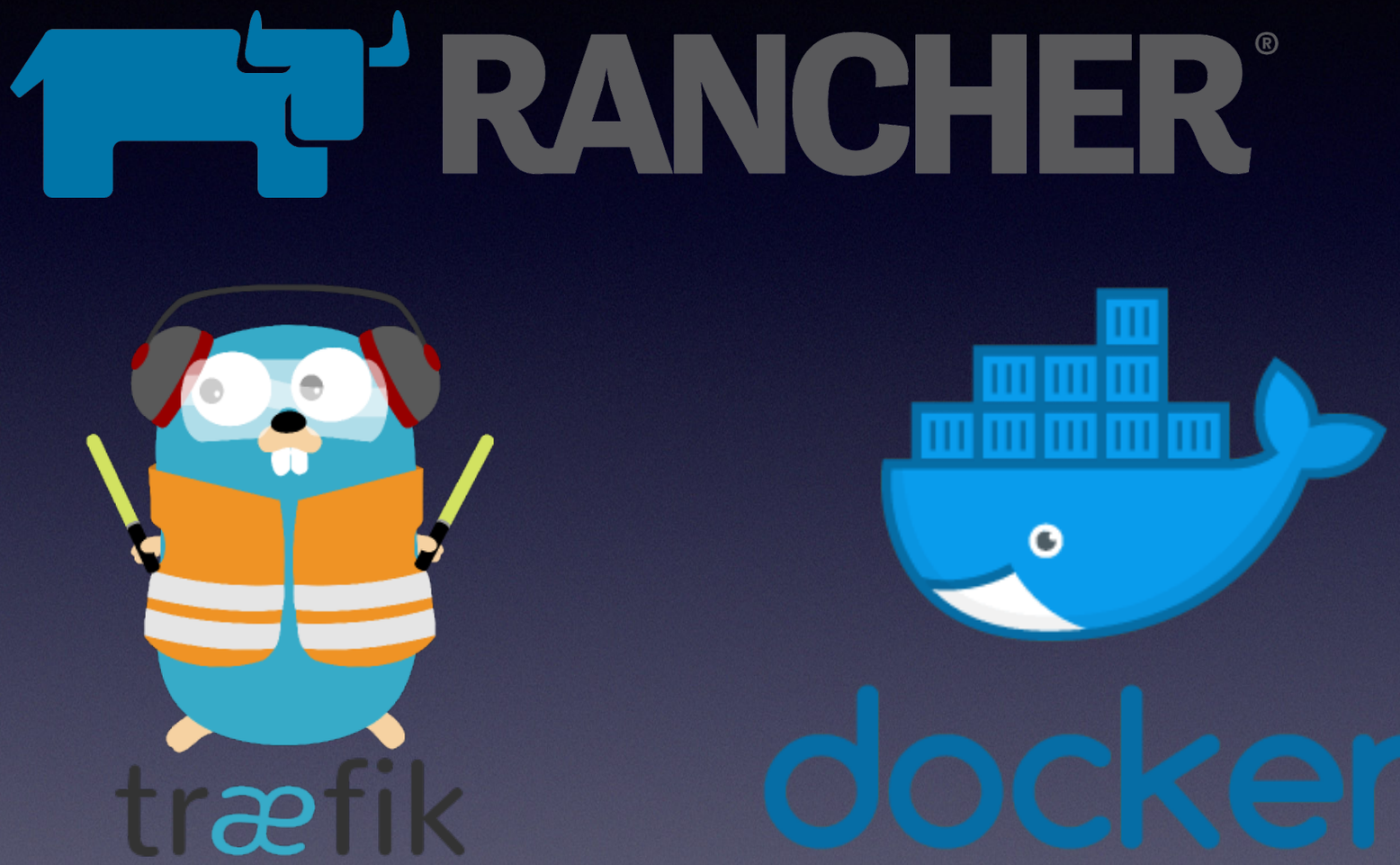
Elastic dockerstalk



http://haah.kr/2019/08/12/php-annotated-monthly-august-2019

PHP 앱을 컨테이너화하는 방법에서는 Nginx + PHP-FPM 조합보다 Apache 웹 서버와 mod_php를 추천하고 있습니다. Nginx가 정적 리소스를 제공하는데 훨씬 유리한 면이 있지만 PHP 앱은 보통 API 서버로 동작하며 정적 리소스는 CDN을 통해 제공하므로, 관리 측면에서 Apache + mod_php가 유리하다는 의견입니다.

가난한 온프라미스 배포 전략


RANCHER®
træfik
docker

RANCHER®

tag-traefik

tag-service

docker

træfik

80

NGINX

9000

phpfpm

socket

```yaml
version: '2'
services:
  app:
    image: docker.yupmin.com/yupmin/some-api:0.0.1
    restart: always
    working_dir: /var/www/html
    environment:
    - ...
    volumes:
    - /var/www/html
    - /etc/localtime:/etc/localtime:ro
    labels:
      io.rancher.container.pull_image: always
      io.rancher.scheduler.affinity:host_label: yupminSomeApi=true
      io.rancher.sidekicks: web
  web:
    image: docker.yupmin.com/yupmin/some-api-web:0.0.1
    restart: always
    working_dir: /var/www/html
    volumes_from:
    - app
    volumes:
    - /etc/localtime:/etc/localtime:ro
    ports:
    - :80
    labels:
      io.rancher.container.pull_image: always
      traefik.backend: yupmin-some-api
      traefik.port: 80
      traefik.frontend.rule: Host:some-api.yupmin.com
      traefik.enable: true
```

Rancher + traefik

RANCHER®

A

B

Rolling update

docker

træfik

NGINX

phpfpm

env

Rancher, Docker 운영시 주의점

- 호스트 볼륨을 Docker 에 마운트 하지 말아야 한다.
  - nginx logs, laravel logs...?
  - docker 가 언제든지 restart 될 수 있음을 숙지 한다.
- 공용 세션, 공용 스토리지는 서비스로 분리되어야 한다.
  - redis, memcached, s3, minio
- 모든 셋팅은 환경변수로 한다.

How to write cloud native and container ready PHP applications

https://withblue.ink/2019/07/24/cloud-native-container-ready-php.html

사람들은 여전히 PHP를 싫어한다면서, “PHP는 섹시하지는 않지만 여전히 웹을 지배합니다.”라고 운을 뗍니다.

PHP가 cloud native하게 개발하는데 문제가 되는 부분은 없으나, …??

(지난 20년 동안의 전통적인 PHP 개발 관행에 위배되는) cloud native PHP application를 설계할 때 중요한 4가지를 지적하고 싶다고 합니다.

1. 로컬 파일 시스템에서 멀어져야 합니다
2. 세션 정보를 Redis에 저장해야 합니다
3. 설정 파일을 환경 변수로 대체해야 합니다
4. built-in 인스톨러나 updater를 사용하면 안됩니다

두번째 항목에서는 꼭 Redis만을 써야 한다는 말은 아니고 세션서버를 분리해야 한다는 의미입니다.

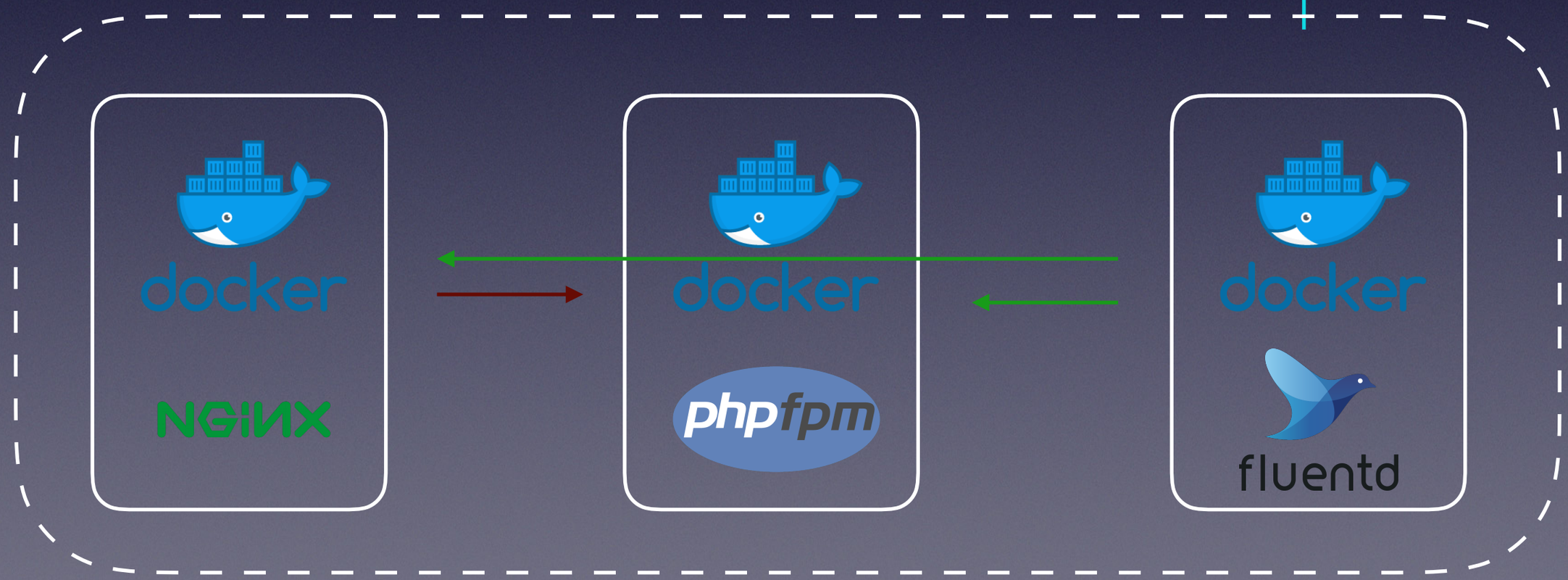
PHP 앱을 컨테이너화하는 방법에서는 Nginx + PHP-FPM 조합보다 Apache 웹 서버와 mod_php를 추천하고 있습니다. Nginx가 정적 리소스를 제공하는데 훨씬 유리한 면이 있지만 PHP 앱은 보통 API 서버로 동작하며 정적 리소스는 CDN을 통해 제공하므로, 관리 측면에서 Apache + mod_php가 유리하다는 의견입니다.

Grafana

fluentd

kibana

elastic



dynimic .env


HashiCorp
Consul

docker

NGINX

docker

phpfpm

Dynamic Request Routing
Service Discovery
Failure Detection

config

dns

control plane

HashiCorp Consul

docker

fluentd

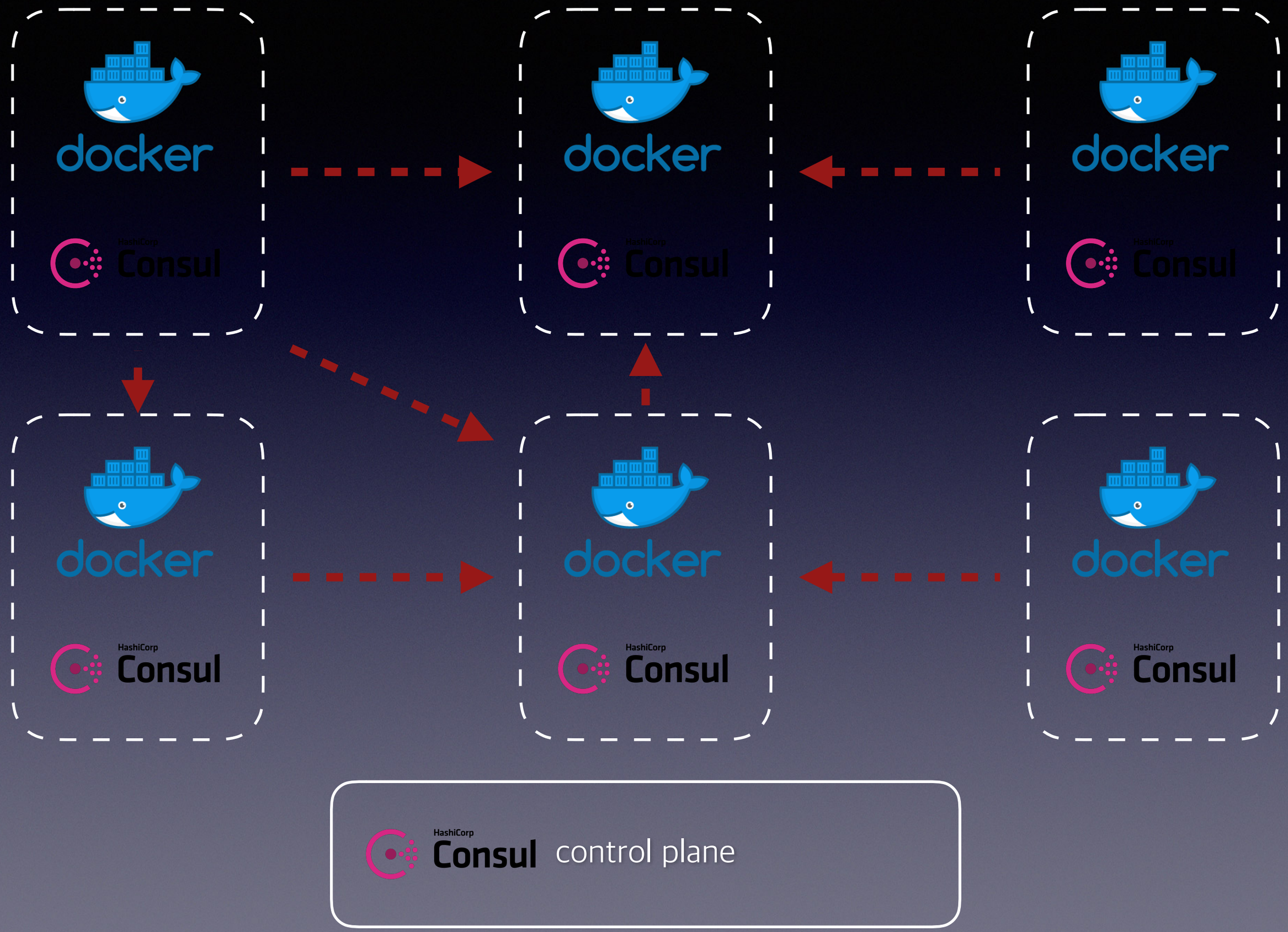
docker

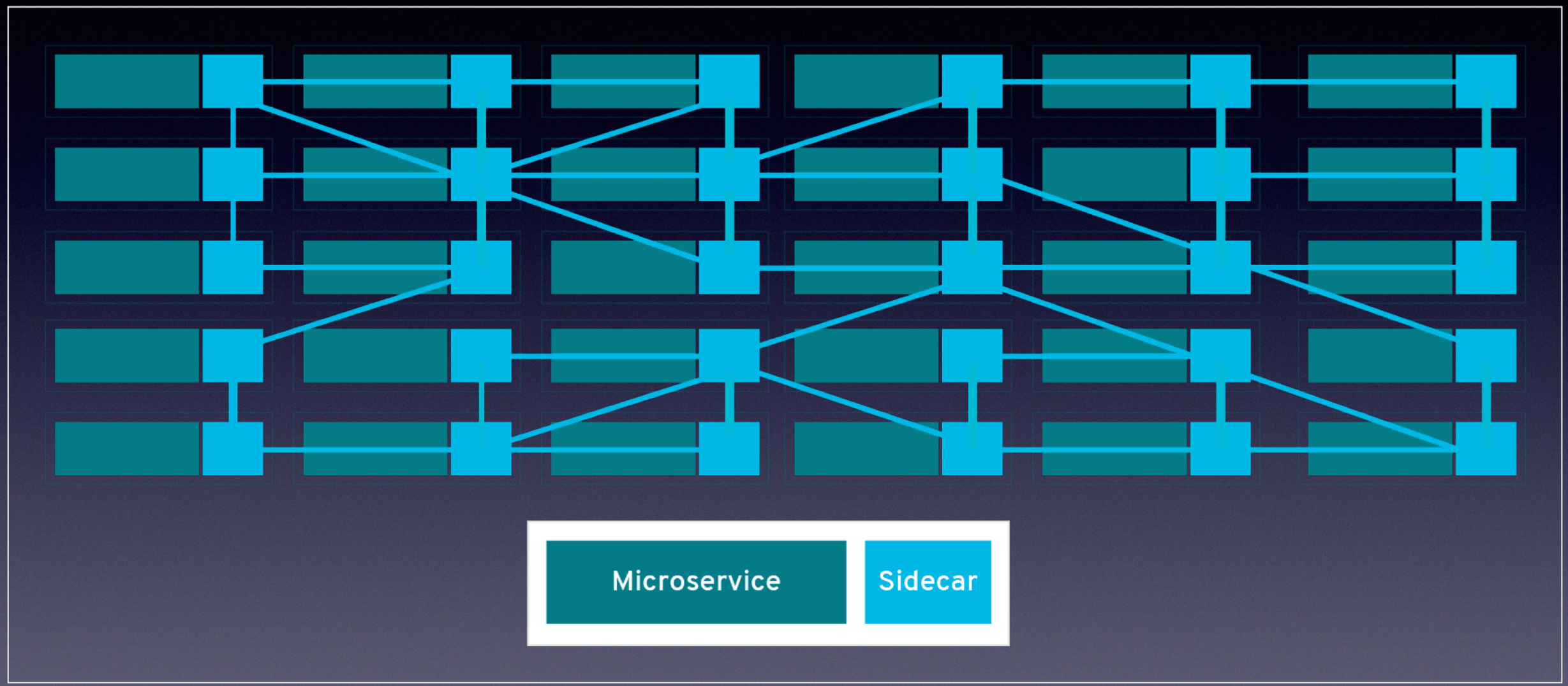
HashiCorp Consul

http request

consul connect

sidecar

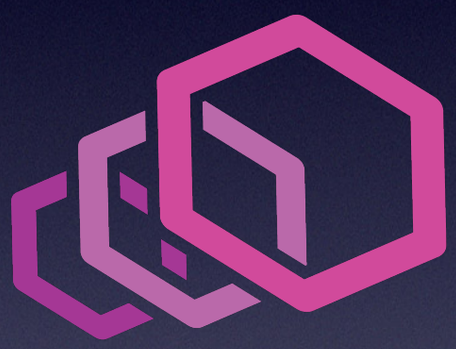

docker
HashiCorp
Consul
docker
HashiCorp
Consul
docker
HashiCorp
Consul
docker
HashiCorp
Consul
docker
HashiCorp
Consul
docker
HashiCorp
Consul
HashiCorp
Consul
control plane


Microservice
Sidecar

Service Mash!!

Service Mesh를 Amazon EKS로 구현하기
– 김세호 솔루션즈 아키텍트(AWS)
https://www.youtube.com/watch?v=d_GXglxEgQE

kubernetes



envoy

Istio

LINKERD

결론

- 프로비져닝, 소스 포함된 도커 빌드로 개발/서비스 운영
- CI/CD 파이프 라인 구축
- 멀티 컨테이너/환경변수 중심 설정 배포 프로세스
- 도커/컨테이너 모니터링은 필수
- rancher, traefik, fluentd, consul 등 조합을 고려

감사합니다.