mijndomein labs

# Running docker in production

Roy Bauweraerts & Erwin de Keijzer

mijndomein

# Hello!

Even voorstellen

```
main( ) {
        printf("hello, world");
}
```

mijndomein

# Hello!

Roy & Erwin

bla bla wie zijn wij ...

@_bauro @erwindekeijzer #labtalks

mijndomein

# Hello!

Roy Bauweraerts

bla bla wie ben ik ...

@_bauro

www.linkedin.com/in/roy-bauweraerts

mijndomein

# Hello!

Erwin de Keijzer

bla bla wie ben ik ...

@erwindekeijzer

www.linkedin.com/in/erwindekeijzer

mijndomein

# md3

## behold the monolith

- eigen ijzer
- 1 release per 4 weken (+ hotfixes)
  - veel nacht releases
- handmatig proces
- releases enkel door Operations

mijndomein

# µd3

behold the distributed monolith

- 3 dedicated aws ec2 instances per microservice
- meerdere releases per week
- niet ontkoppeld
- releases door Developers

mijndomein

# Het doel!

## Waarom eigenlijk?

Een platform opzetten waarbij het mogelijk is om, met hoge snelheid en zekerheid, bestaande functionaliteit te wijzigen of nieuwe functionaliteit toe te voegen, zonder daarbij de beschikbaarheid van de producten voor klanten te compromitteren.

mijndomein

# Docker

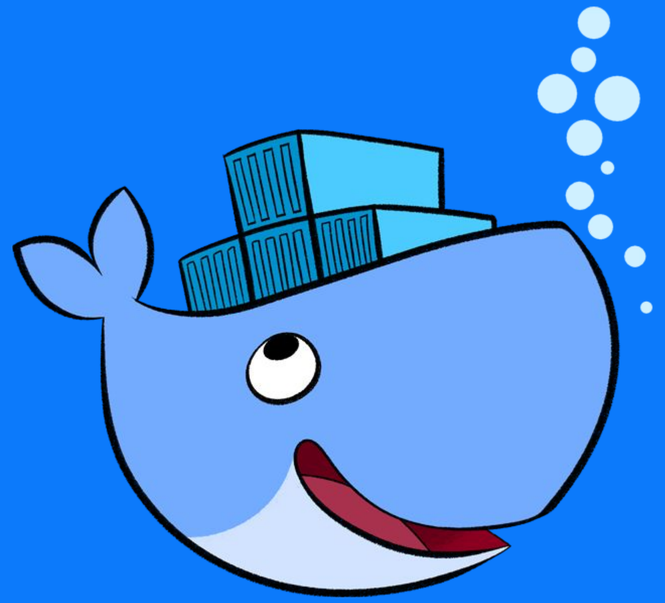
## wat uitleg

so,
what is
this?

mijndomein

# Docker

## In het kort



Docker containers wrap a piece of software in a complete filesystem that contains everything needed to run:

code, runtime, system tools, system libraries –

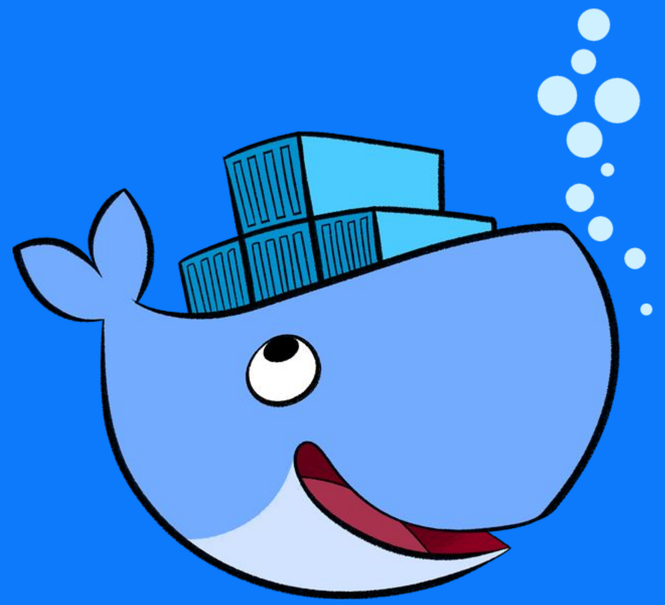
anything that can be installed on a server.

This guarantees that the software will always run the same, regardless of its environment.

mijndomein

# Docker

## In het kort



Docker containers wrap a piece of software in a complete filesystem that contains everything needed to run:

**code**, runtime, system tools, system libraries –

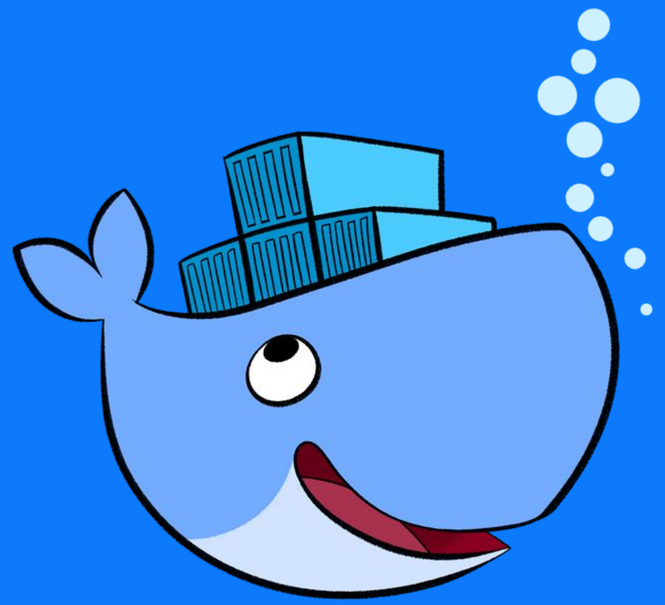
anything that can be installed on a server.

This guarantees that the software will always run the same, regardless of its environment.

mijndomein

# Docker

## In het kort



Docker containers wrap a piece of software in a complete filesystem that contains everything needed to run:

code, **runtime**, system tools, system libraries –

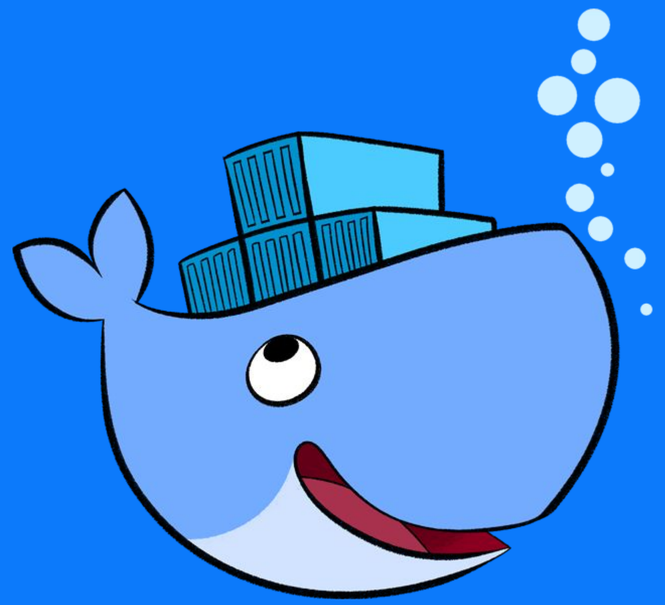
anything that can be installed on a server.

This guarantees that the software will always run the same, regardless of its environment.

mijndomein

# Docker

## In het kort



Docker containers wrap a piece of software in a complete filesystem that contains everything needed to run:

code, runtime, **system tools**, system libraries –

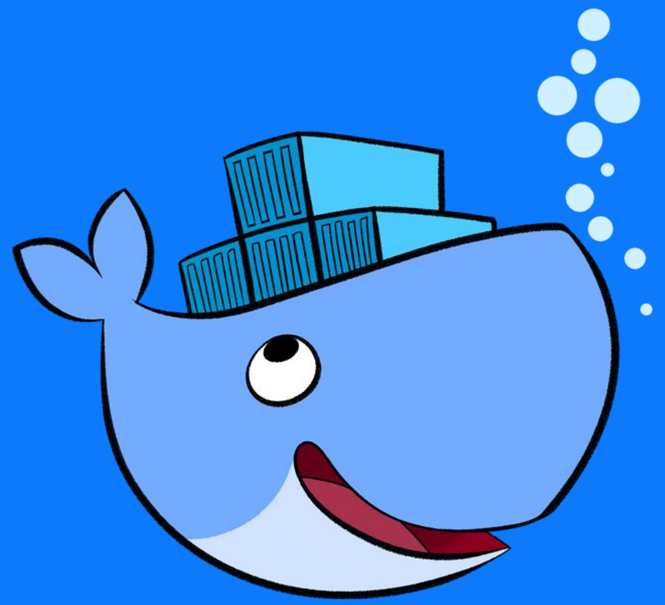
anything that can be installed on a server.

This guarantees that the software will always run the same, regardless of its environment.

mijndomein

# Docker

## In het kort



Docker containers wrap a piece of software in a complete filesystem that contains everything needed to run:

code, runtime, system tools, **system libraries** –

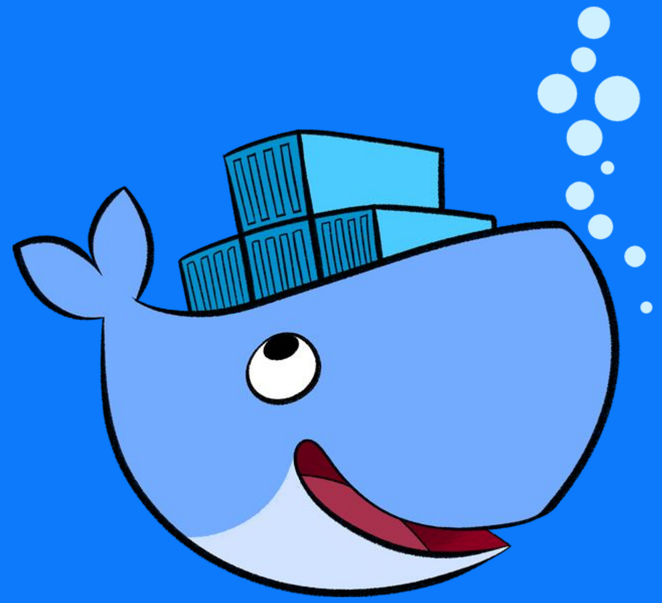
anything that can be installed on a server.

This guarantees that the software will always run the same, regardless of its environment.

mijndomein

# Docker

## In het kort



Docker containers wrap a piece of software in a complete filesystem that contains everything needed to run:

code, runtime, system tools, system libraries –

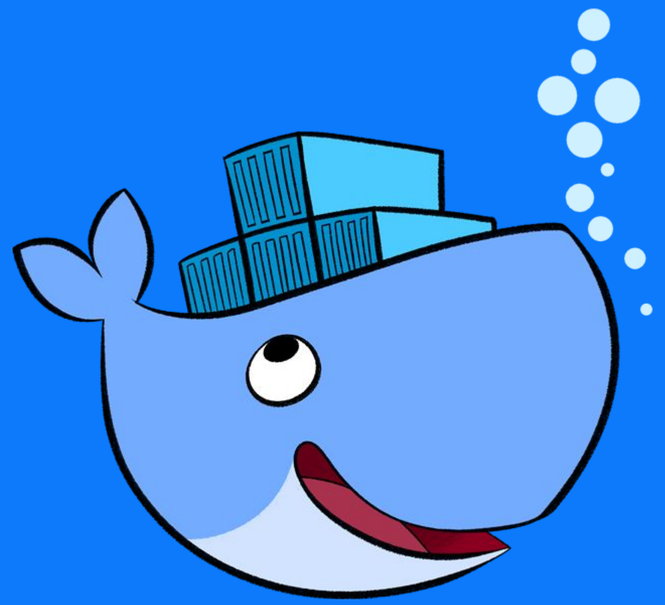
anything that can be installed on a server.

This guarantees that the software **will always run the same**, regardless of its environment.

mijndomein

# Docker

## In het kort



Docker containers wrap a piece of software in a complete filesystem that contains everything needed to run:

code, runtime, system tools, system libraries –

anything that can be installed on a server.

This guarantees that the software will always run the same, **regardless of its environment**.

mijndomein

# Challenges

Do you accept?


- docker containers draaien
- environment consistency & configuration
- service discovery
- logging
- monitoring
- request routing
- updates zonder downtime

mijndomein

# Challenge

## #1 Container draaien

Hoe zorg je ervoor dat je met minimale moeite containers kan:

- toevoegen
- verwijderen
- updaten
- schalen

zonder dat bezoekers hier iets van merken?

mijndomein

# Challenge

## #1 Container draaien

Kubernetes

“Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications.”

mijndomein

# Challenge

## #1 Container draaien


Kubernetes

Veel bewegende onderdelen & ingewikkelde bootstrap

mijndomein

# Challenge

## #1 Container draaien

Nomad

“A cluster manager and scheduler designed for microservices and batch workloads.
Nomad is distributed, highly available, and scales to thousands of nodes spanning multiple datacenters and regions.”

mijndomein

# Challenge

## #1 Container draaien

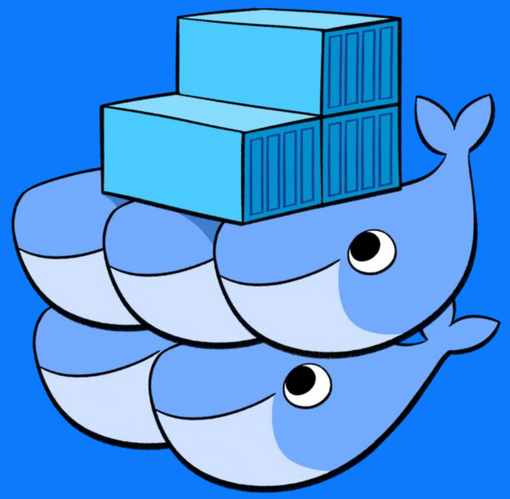

Nomad

Geen rolling updates op basis van healthchecks

mijndomein

# Challenge

## #1 Container draaien



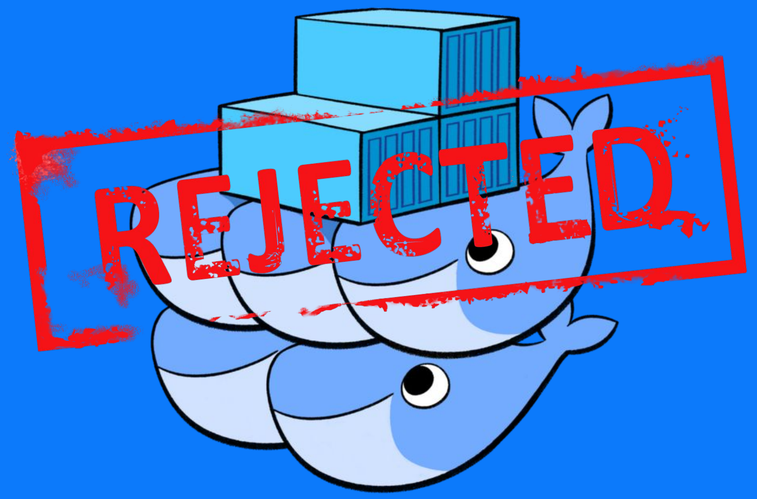
Docker swarm mode

“Docker Engine 1.12 includes swarm mode for natively managing a cluster of Docker Engines called a swarm. Use the Docker CLI to create a swarm, deploy application services to a swarm, and manage swarm behavior.”

mijndomein

# Challenge

## #1 Container draaien



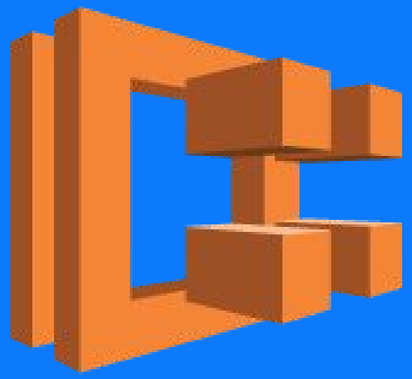

Docker swarm mode

Niet stabiel

mijndomein

# Challenge

## #1 Container draaien



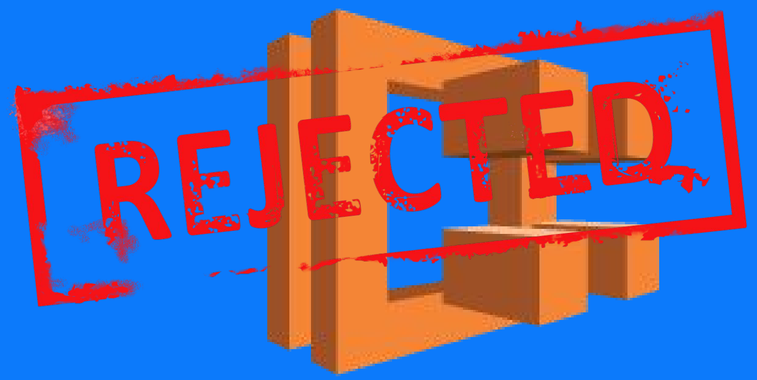
### Amazon ECS

“Amazon EC2 Container Service (Amazon ECS) is a highly scalable, fast, container management service that makes it easy to run, stop, and manage Docker containers on a cluster of Amazon Elastic Compute Cloud (Amazon EC2) instances.”

mijndomein

# Challenge

## #1 Container draaien




Amazon ECS

Vendor lock in

mijndomein

# Challenge

## #1 Container draaien

DC/OS

“DC/OS is an enterprise grade datacenter-scale operating system,
providing a single platform for running containers, big data, and distributed apps in production.”

mijndomein

# Challenge

## #1 Container draaien

mijndomein

# Challenge

## #1 Container draaien

Voordelen:

- (relatief) simpele bootstrap
- CRUD web interface
- goede logging opties
- healthchecks voor rolling updates

Nadelen:

- veel bewegende onderdelen
- distrubeted state
- geen drain-mode
- geen system-jobs
- geen native consul integratie
- web interface is niet perfect

mijndomein

# Challenge

## #1 Container draaien

mijndomein

# Challenges

Do you accept?


- docker containers draaien
- environment consistency & configuration
- service discovery
- logging
- monitoring
- request routing
- updates zonder downtime

mijndomein

# Challenge

## #2 Environment

Hoe zorg je ervoor dat je code hetzelfde resultaat geeft:

Hier

En hier

mijndomein

# Challenge

## #2 Environment

- 1 artifact gebruiken voor al je omgevingen
- artifact bundelt alles wat nodig is om je code te draaien:
  - os
  - libraries
  - plugins
  - tooling
- configuratie wordt tijdens runtime geinjecteerd

mijndomein

# Challenge

## #2 Environment

- 1 artifact gebruiken voor al je omgevingen
- artifact bundelt alles wat nodig is om je code te draaien:
  - os
  - libraries
  - plugins
  - tooling
- configuratie wordt tijdens runtime geinjecteerd

mijndomein

# Challenge

## #2 Environment

- 1 artifact gebruiken voor al je omgevingen
- artifact bundelt alles wat nodig is om je code te draaien:
  - os
  - libraries
  - plugins
  - tooling
- configuratie wordt tijdens runtime geïnjecteerd

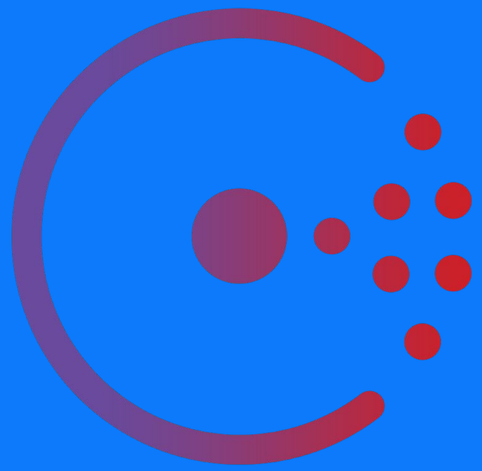
} docker

} consul-template

mijndomein

# Challenge

## #2 Environment



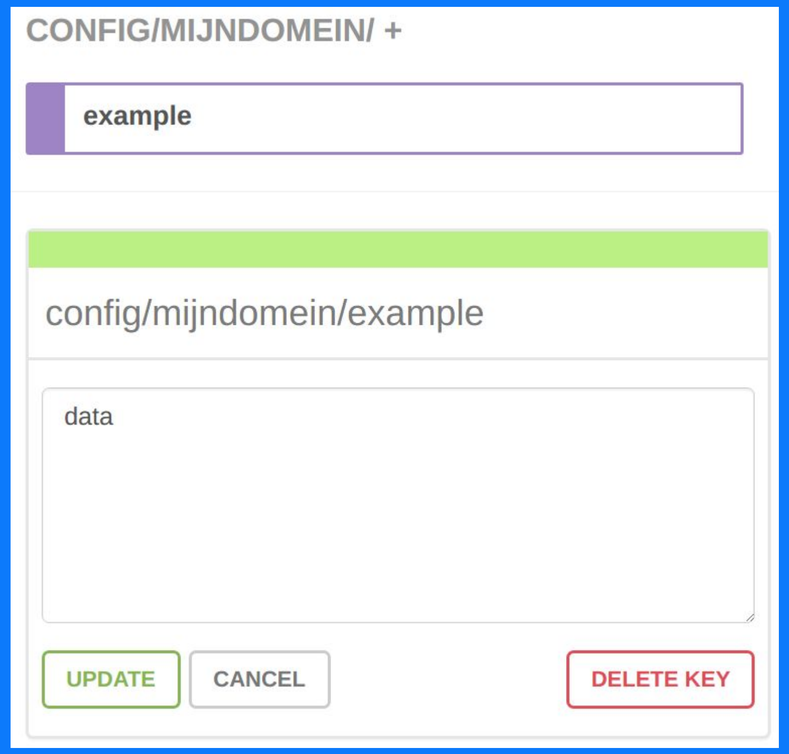
Consul

- Service discovery
- Failure detection
- Multi datacenter
- Key/Value storage

mijndomein

# Challenge

## #2 Environment

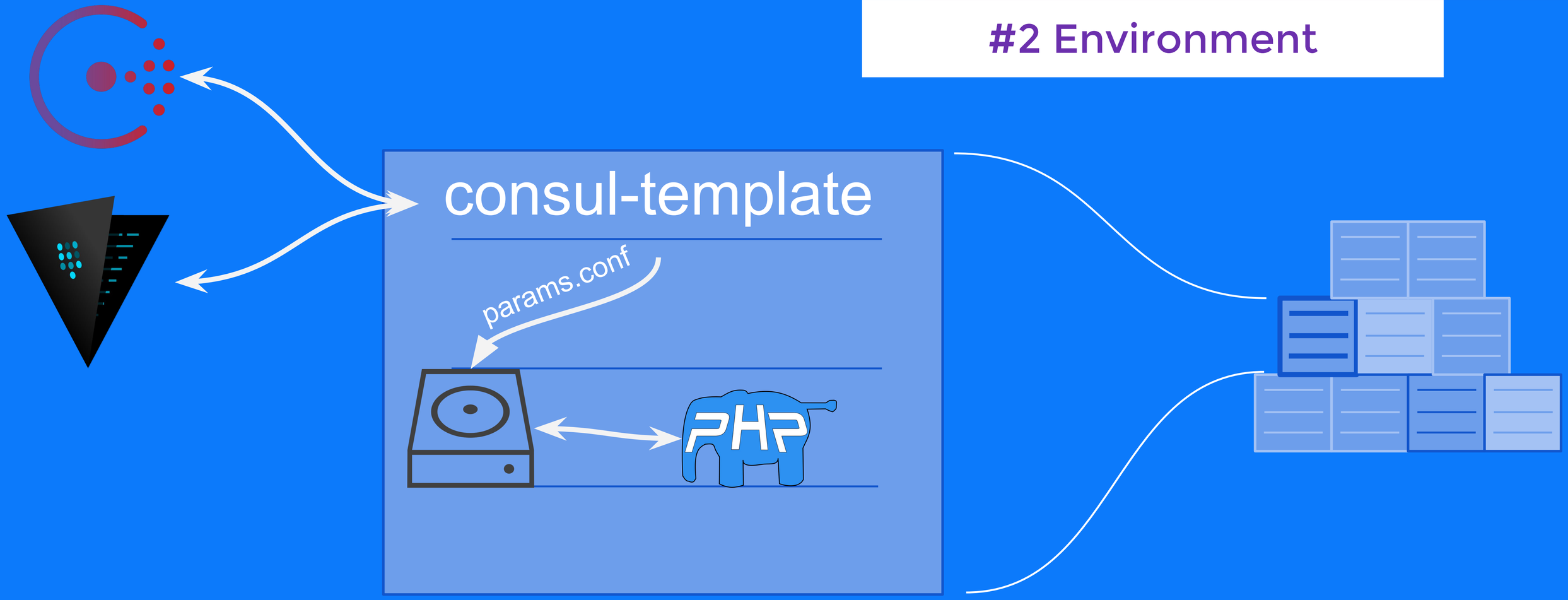



```
➪ cat parameters.yml.ctmpl
---
{{ tree "config/mijndomein" | explode | toYAML }}

➪ consul-template -consul consul.service.consul:8500
-once -template "parameters.yml.ctmpl:parameters.yml"

➪ cat parameters.yml
---
example: data
```

mijndomein

# Challenge

## #2 Environment



mijndomein

# Challenge

## #2 Environment

mijndomein

# Challenges

Do you accept?

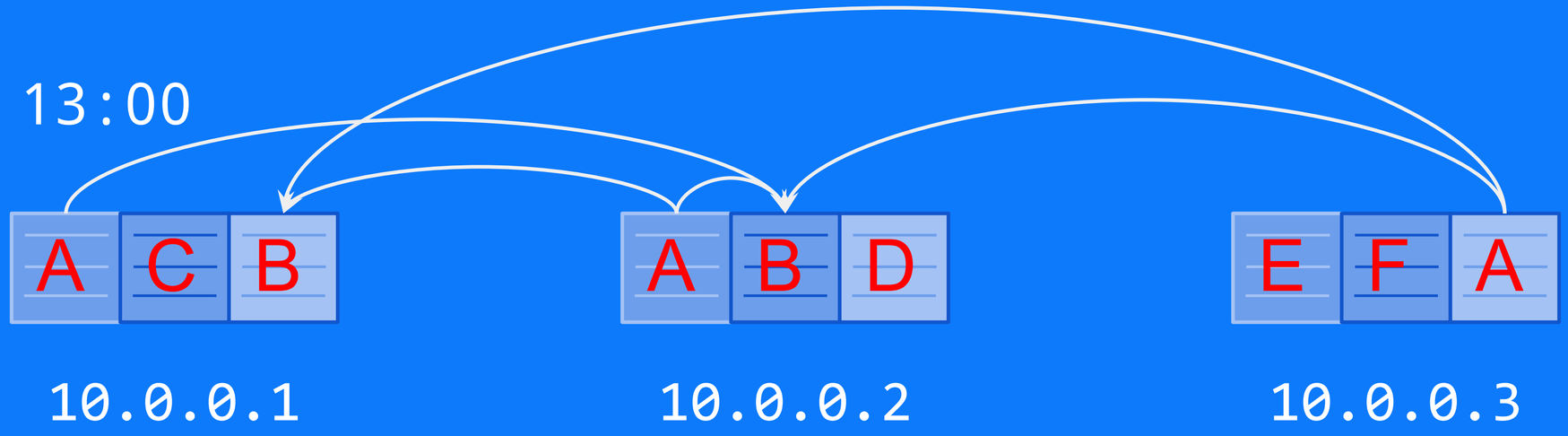

- docker containers draaien
- environment consistency & configuration
- service discovery
- logging
- monitoring
- request routing
- updates zonder downtime

mijndomein

# Challenge

## #3 Service discovery

Hoe zorg je ervoor dat containers met elkaar kunnen communiceren in een continu veranderend landschap:



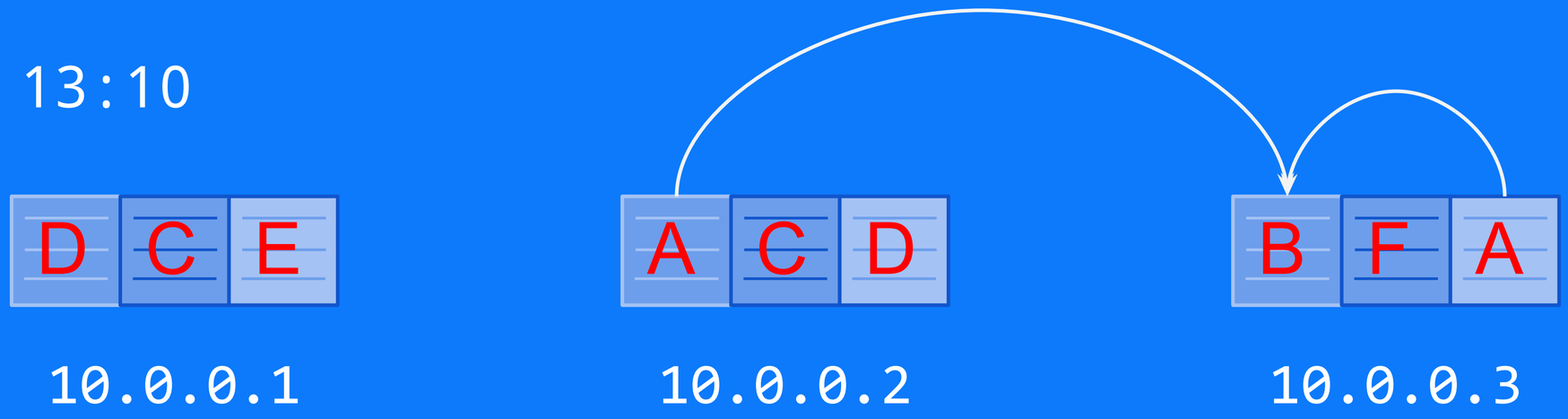
mijndomein

# Challenge

## #3 Service discovery

Hoe zorg je ervoor dat containers met elkaar kunnen communiceren in een continu veranderend landschap:

13:10

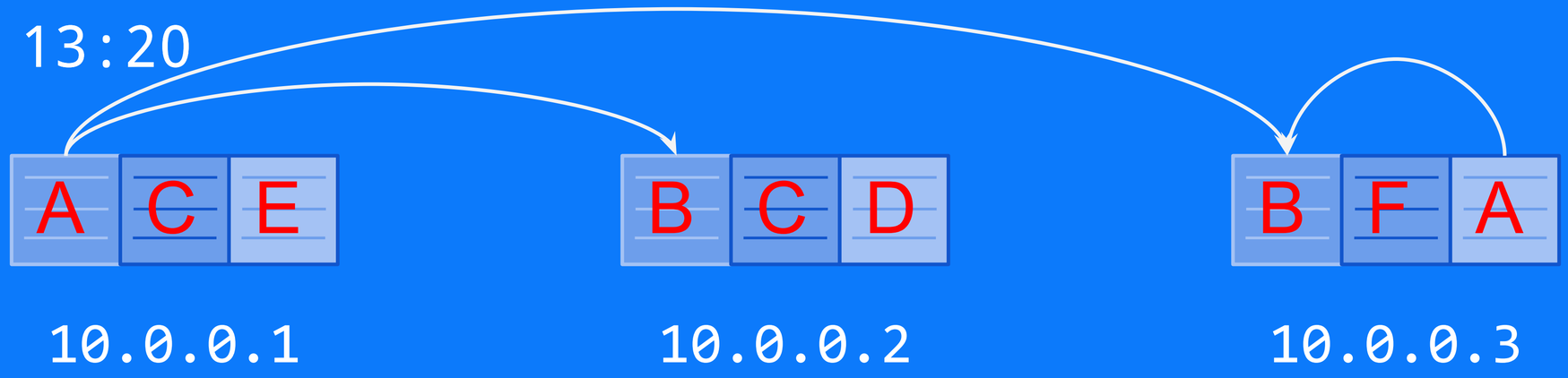


mijndomein

# Challenge

## #3 Service discovery

Hoe zorg je ervoor dat containers met elkaar kunnen communiceren in een continu veranderend landschap:



mijndomein

# Challenge

## #3 Service discovery


### Mesos DNS

We hebben ook services buiten mesos

### DC/OS service ports

Van buitenaf is ook ip registratie nodig

### Consul DNS

DC/OS kan niet communiceren met consul

mijndomein

# Challenge

## #3 Service discovery


Mesos consul

“Mesos to Consul bridge for service discovery”

mijndomein

# Challenge

## #3 Service discovery

- Kijkt naar mesos
- Registreert tasks als applicationid.service.consul
  - (kan overschreven worden met marathon labels)
- Kan consul healthchecks registreren in consul
- Update op een vast interval
  - niet ideaal, compromis tussen consistency en performance

mijndomein

# Challenge

## #3 Service discovery

mijndomein

# Challenges

Do you accept?


- docker containers draaien
- environment consistency & configuration
- service discovery
- logging
- monitoring
- request routing
- updates zonder downtime

mijndomein

# Challenge

## #4 Logging


Hoe zorg je ervoor dat je met minimale moeite kan weten wat er gebeurt:

- op applicatie niveau
- op domain niveau

mijndomein

# Challenge

## #4 Logging

Applicatie

stdout inzichtelijk via webinterface

mijndomein

# Challenge

## #4 Logging

Domain

Events via rabbitmq en opgeslagen in elasticsearch

mijndomein

# Challenge

## #4 Logging

mijndomein

# Challenges

Do you accept?


- docker containers draaien
- environment consistency & configuration
- service discovery
- logging
- monitoring
- request routing
- updates zonder downtime

mijndomein

# Challenge

## #5 Monitoring

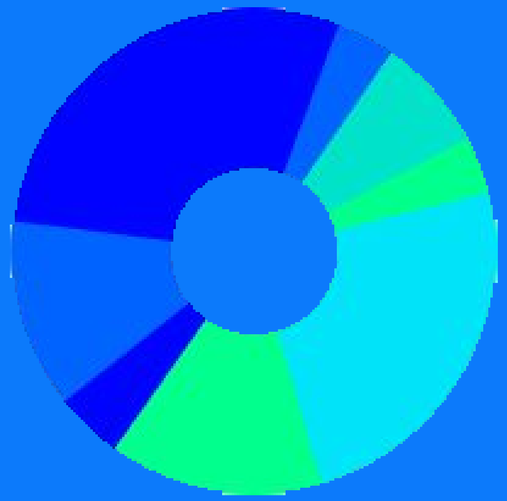
Hoe zorg je ervoor dat containers door automatische checks healthy blijven draaien?:

- marathon checks
- alerting met datadog

mijndomein

# Challenge

## #5 Monitoring



## Marathon

Health Checks

```
[
  {
    "protocol": "COMMAND",
    "command": {
      "value": "redis-cli ping"
    },
    "gracePeriodSeconds": 300,
    "intervalSeconds": 60,
    "timeoutSeconds": 20,
    "maxConsecutiveFailures": 3,
    "ignoreHttp1xx": false
  }
]
```

mijndomein

# Challenge

## #5 Monitoring


Datadog

- Trends op basis van opgeslagen metrics
- Alerting dmv ingestelde limieten

mijndomein

# Challenge

## #5 Monitoring

mijndomein

# Challenges

Do you accept?


- docker containers draaien
- environment consistency & configuration
- service discovery
- logging
- monitoring
- request routing
- updates zonder downtime

mijndomein

# Challenge

## #6 Request routing


Hoe zorg je ervoor dat requests bij de goede container terecht komen:

van hier

AWS
ELB / ALB

naar daar

mijndomein

# Challenge

## #6 Request routing

```
GET / HTTP/1.1
Host: www.mijndomein.nl
```

AWS
ELB / ALB

10.0.0.1:32001

DCOS
Agents

mijndomein

# Challenge

## #6 Request routing

```
GET /producten HTTP/1.1
Host: www.mijndomein.nl
```

AWS
ELB / ALB

10.0.0.2:32003

DCOS
Agents

mijndomein

# Challenge

## #6 Request routing

```
GET /login HTTP/1.1
Host: auth.mijndomein.nl
```

AWS
ELB / ALB

10.0.0.3:32005

DCOS
Agents

mijndomein

# Challenge

## #6 Request routing


containers registreren bij AWS ALB
Complex en gevoelig voor fouten

Statische proxy (nginx, apache2, haproxy)
Simpel maar veel handmatig werk

Dynamische proxy (fabio/traefik)
Vrij simpel maar beperkt

Alle containers op alle hosts laten draaien
Enorm veel overhead en weinig dynamisch

mijndomein

# Challenge

## #6 Request routing


Traefik

“Træfɪk is a modern HTTP reverse proxy and load balancer made to deploy microservices with ease.”

mijndomein

# Challenge

## #6 Request routing

træfik

Host: auth.mijndomein.nl

Host: www.mijndomein.nl

AWS
ELB / ALB

nginx proxy

GET / HTTP/1.1

GET /producten HTTP/1.1

DCOS
Agents

mijndomein

# Challenge

## #6 Request routing

mijndomein

# Challenges

Do you accept?


- docker containers draaien
- environment consistency & configuration
- service discovery
- logging
- monitoring
- request routing
- updates zonder downtime

mijndomein

# Challenge

## #7 Rolling updates

Hoe zorg je ervoor dat je kan updaten zonder dat klanten er iets van merken?:

- applicatie
- servers

mijndomein

# Challenge

## #7 Rolling updates

Applicatie

mijndomein

# Challenge

## #7 Rolling updates

mijndomein

# Challenge

## #7 Rolling updates

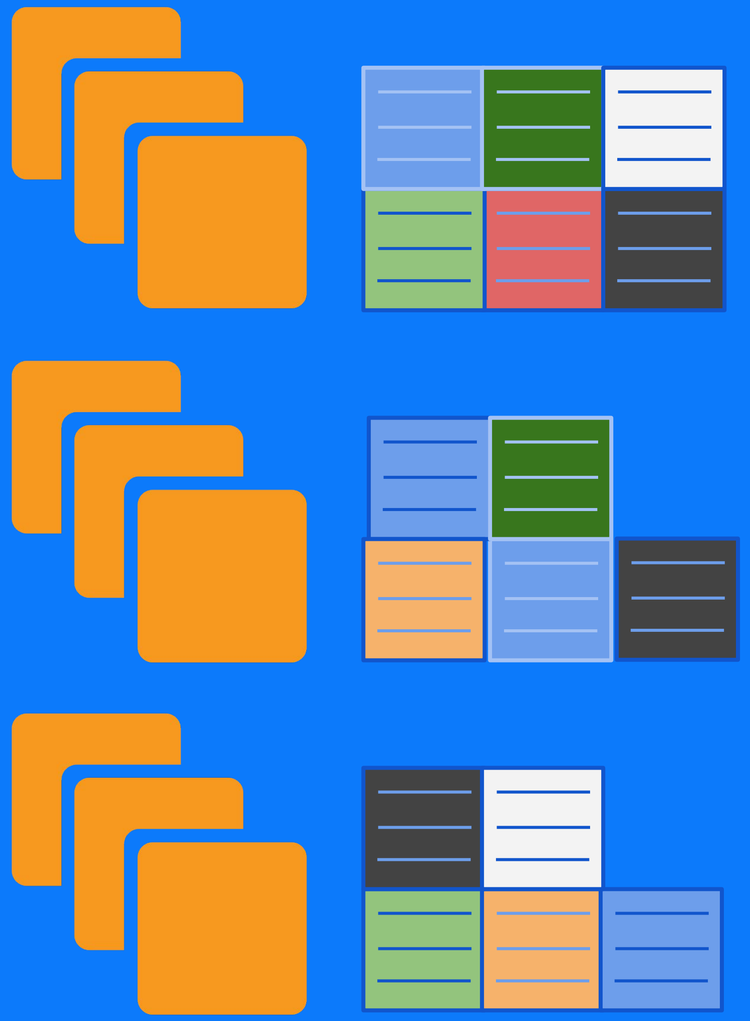
Servers

mijndomein

# Challenge

## #7 Rolling updates

dcos-agents



mijndomein

# Challenge

## #7 Rolling updates

dcos-agents

1a

1b

1c

mijndomein

# Challenge

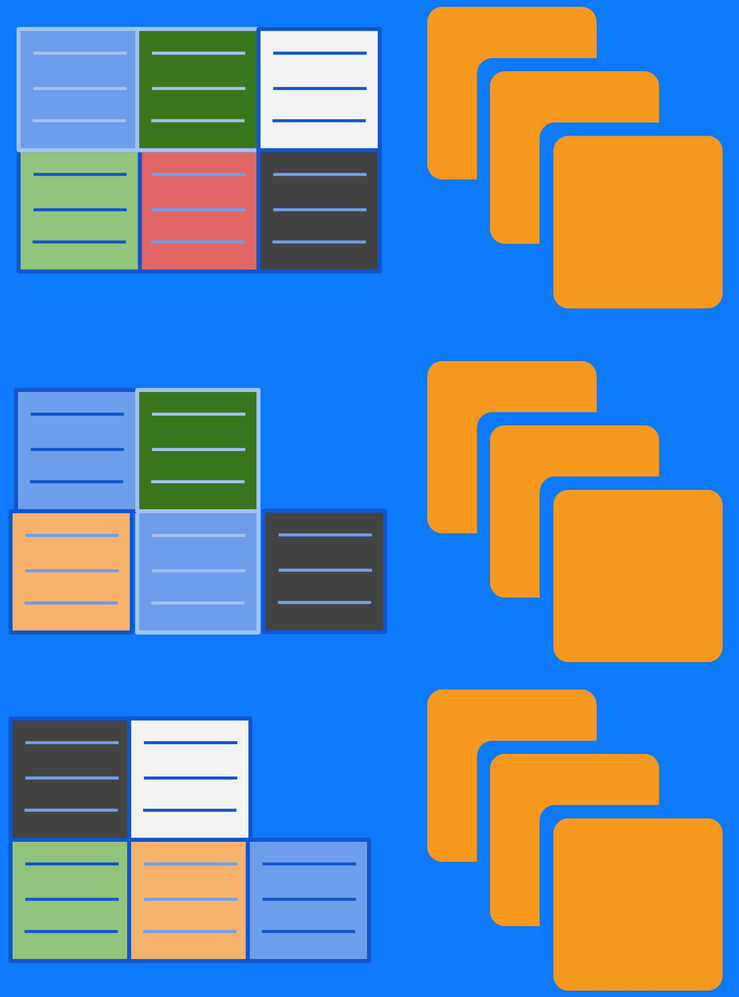
## #7 Rolling updates

dcos-agents

1a

1b

1c



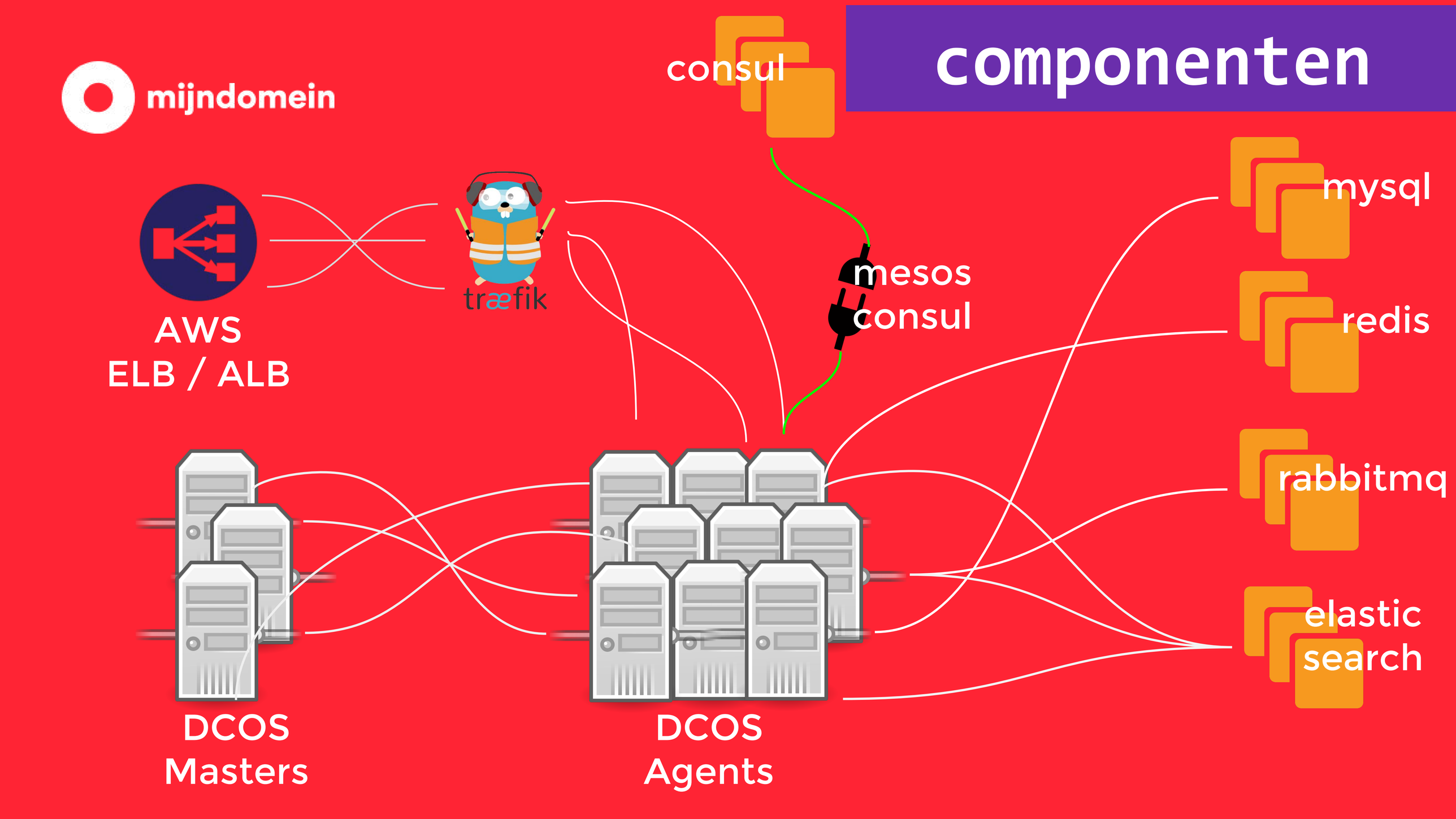
mijndomein

# Challenges

Do you accept?


- docker containers draaien
- environment consistency & configuration
- service discovery
- logging
- monitoring
- request routing
- updates zonder downtime

# componenten



mijndomein

# Bye bye!

That's all folks

https://github.com/mijndomein/docker-in-production-talk

@_bauro @erwindekeijzer #labtalks