

ZEND SERVER IN DOCKER SWARM: DEPLOYMENT AND MORE

Jan Burkl
Solution Consulting Manager
Rogue Wave Software

ZendCon 2017, Las Vegas, October 26th 2017



Load



Particulum Mobile

Partcile collisions

Requests to backend: 8770
Responses from backend: 7883

Dead Partciles

Responses w/o recipients: 1755

PHP Containers

6d6db0d5a2fa	666
d4f8e73eb48e	463
f57a871e5e8b	365
016140aae2d3	364
e501c4fe2126	365
ff40afc93a11	365
72234b6b4497	364
68624300be27	365



Statistics

Session Id:

Collision Count: 8216

Redis, total

Collision Count: 365

Session

The background features a series of thick, light blue diagonal stripes of varying lengths, creating a dynamic, layered effect against a white background.

DOCKER

DOCKERFILE (PROD)

```
FROM php:7.0-fpm

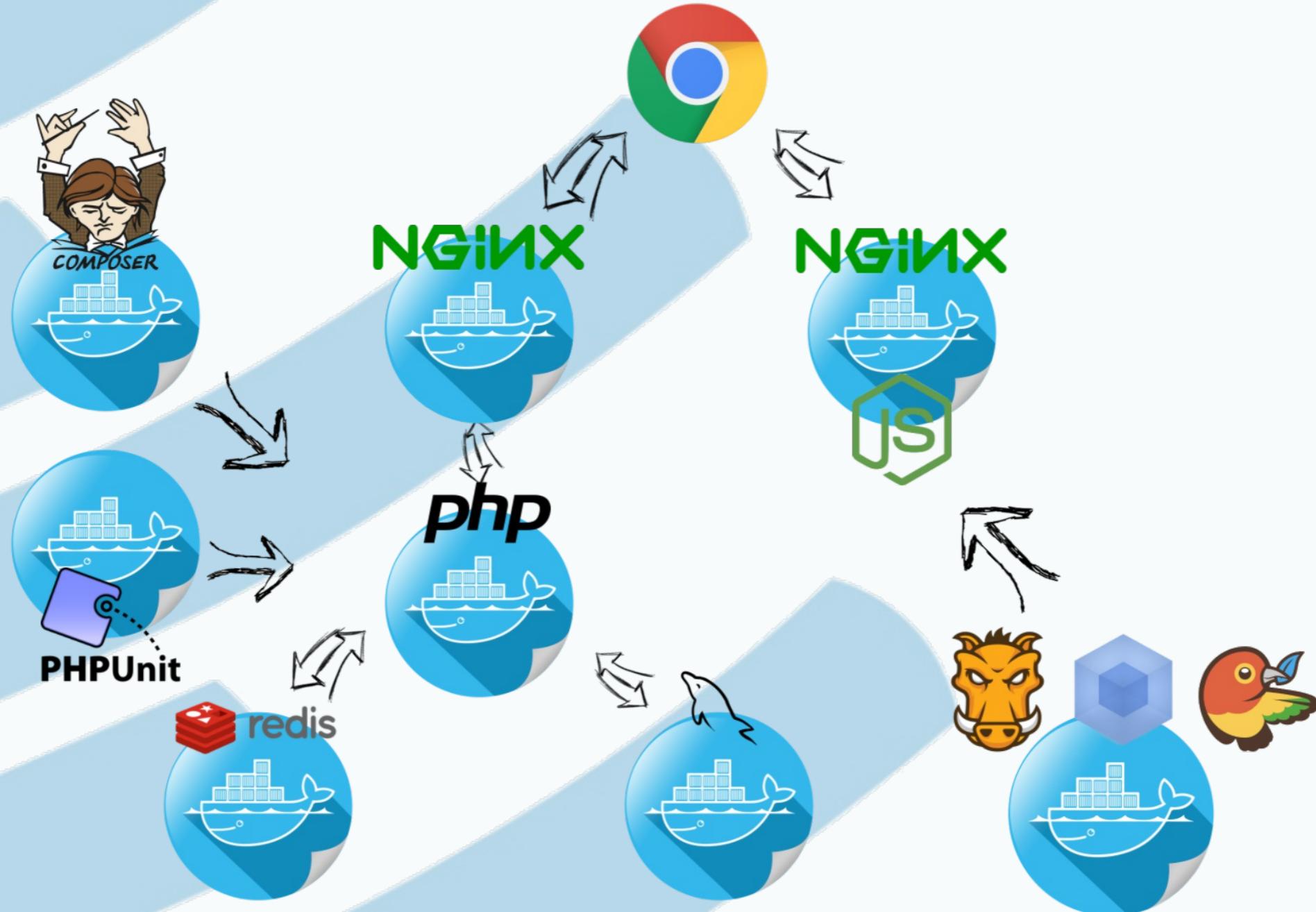
RUN apt-get update && apt-get install -y git libcurl4-gnutls-dev \
&& git clone -b php7 https://github.com/phpredis/phredis \
&& docker-php-ext-install redis \
&& apt-get autoremove && apt-get autoclean \
&& rm -rf /var/lib/apt/lists/*

#-----
# App sources
#-----
COPY . /app/
```

DOCKERFILE (DEV)

```
FROM janatzend/particulum-mobile-backend

WORKDIR /
ADD http://www.zend.com/en/download/4843?start=true .
RUN \
    echo zend_extension=/zend-debugger/ZendDebugger-lin
    echo zend_debugger.allow_from_all=1 >> /usr/local/etc/
```



REDIS CONTAINER

```
$ docker run -d --name redis redis:3.2.7-alpine
```

PHP CONTAINER

```
$ docker run -d --name php \
-v "$PWD:/app" \
--link redis:redis \
particulummobile-dev/backend
```

NGINX CONTAINER

```
$ docker run -d --name nginx \
-v "$PWD:/app" \
-v "$PWD/etc/nginx.site.conf:/etc/nginx/conf.d/default.conf" \
--link php:php \
-p 8888:80 \
particulummobile-dev/backend-nginx
```

DOCKER COMPOSE

```
version: "3"

services:

#-----
# Nginx
#-----
nginx:
  build:
    context: "${PROJECTS_PATH}/${BACKEND_APP_DIR}"
    dockerfile: Dockerfile.nginx
    image: particulummobile-dev/backend-nginx
```

SCALING

```
$ docker-compose scale php=5
```

Load Balancer?

DOCKER SWARM (MODE)

“Current versions of Docker include 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.”

CLOUD.DOCKER.COM

- Beta
- Service Provider
 - AWS
 - Azure
 - On Premise

The background features a series of thick, light blue diagonal stripes of varying lengths, creating a dynamic, layered effect against a white background.

PLAY-WITH-DOCKER.COM

MY LOCAL SETUP

- VirtualBox
- Vagrant
- 3 Nodes
- Ubuntu 16.04

INIT / MANAGER

```
$ docker swarm init --advertise-addr 192.168.99.121
```

Swarm initialized: current node (ww4nime7fbh9twrzzl7o2t

To add a worker to this swarm, run the following command:

```
docker swarm join \
--token SWMTKN-1-2aps743f7wzjz6xp3vnlicpkim9ge2y
192.168.33.101:2377
```

To add a manager to this swarm, run 'docker swarm join-token'

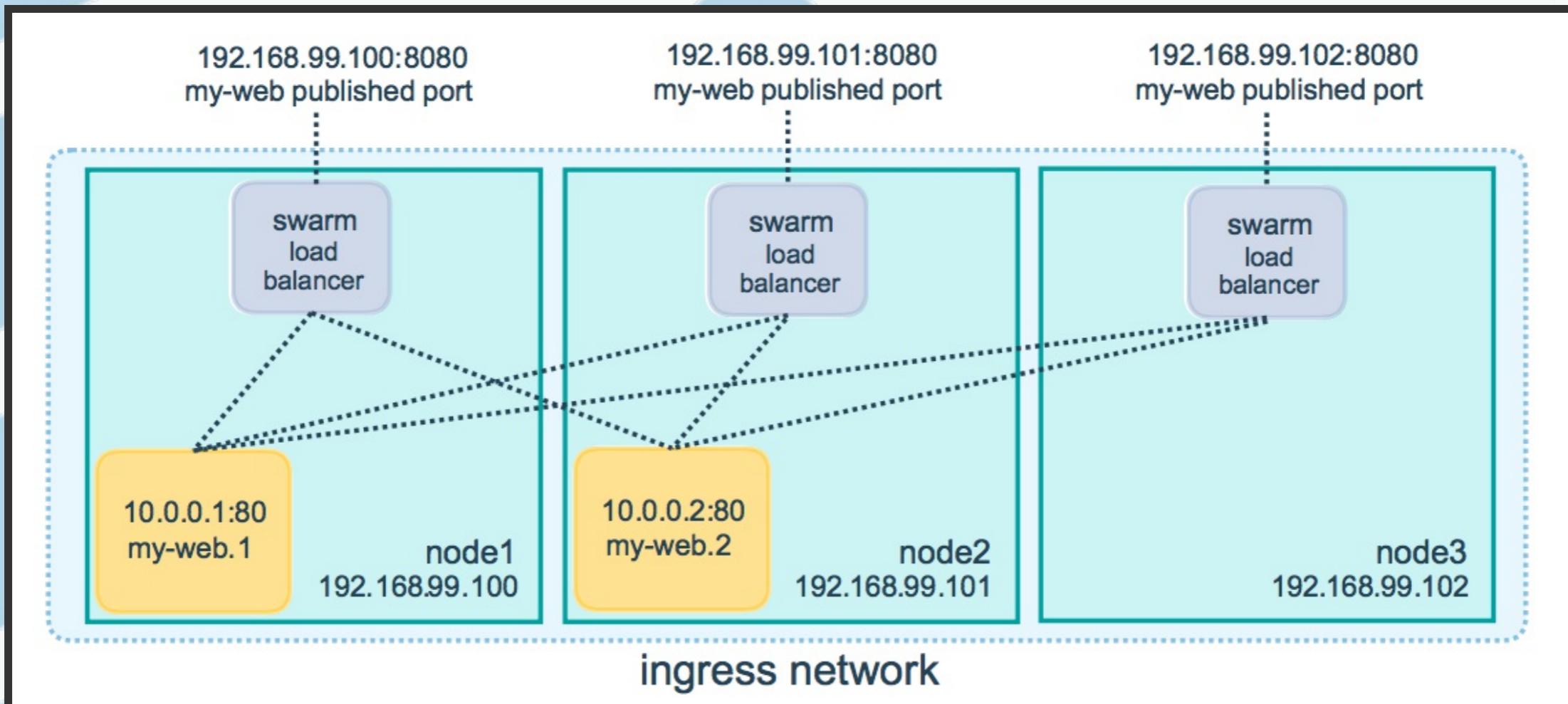
The background features a series of thick, light blue diagonal stripes of varying lengths, creating a sense of depth and motion. The stripes are set against a white background.

NETWORKING

ROUTING MESH

“The routing mesh enables each node in the swarm to accept connections on published ports for any service running in the swarm, even if there’s no task running on the node.”

INGRESS NETWORK



<https://docs.docker.com/engine/swarm/ingress/>

The background features a series of thick, light blue diagonal stripes of varying lengths, creating a dynamic, layered effect against a white background.

SERVICES

● machine4
manager
15G free

● machine5
worker
15G free

● machine7
worker
15G free

● machine8
worker
15G free

EXAMPLE:

DOCKER SWARM VISUALIZER

● busybox

tag : latest

cmd : bash

updated : 7/9 18:28

bdaad0d1db393b62fec5e5d19a66

state : starting

● busybox

tag : latest

cmd : bash

updated : 7/9 18:28

c5fdb012ed62764f03ba99068182430

state : failed

<https://github.com/dockersamples/docker-swarm-visualizer>

● rabbitmq

tag : 3.6.5

updated : 5/9 23:38

9328ab4df3cc7218fb44dd42196aab7

state : running

● busybox

tag : latest

cmd : sleep,444

updated : 7/9 18:27

2bf67c98d4310e77a053784f9d03a04

state : running

● rabbitmq

tag : 3.6.5

updated : 7/9 18:21

3c7510b1f9aaf77c730067f74451e05

state : running

● rabbitmq

tag : 3.6.5

updated : 7/9 18:21

2ddd5a5232284b228299835b6899b4

state : running

CREATE SERVICE

```
$ docker service create \  
--name=viz \  
--publish=8080:8080/tcp \  
--constraint=node.role==manager \  
--mount=type=bind,src=/var/run/docker.sock,dst=/var/ \  
dockersamples/visualizer
```

BUILDING IMAGES

self-contained

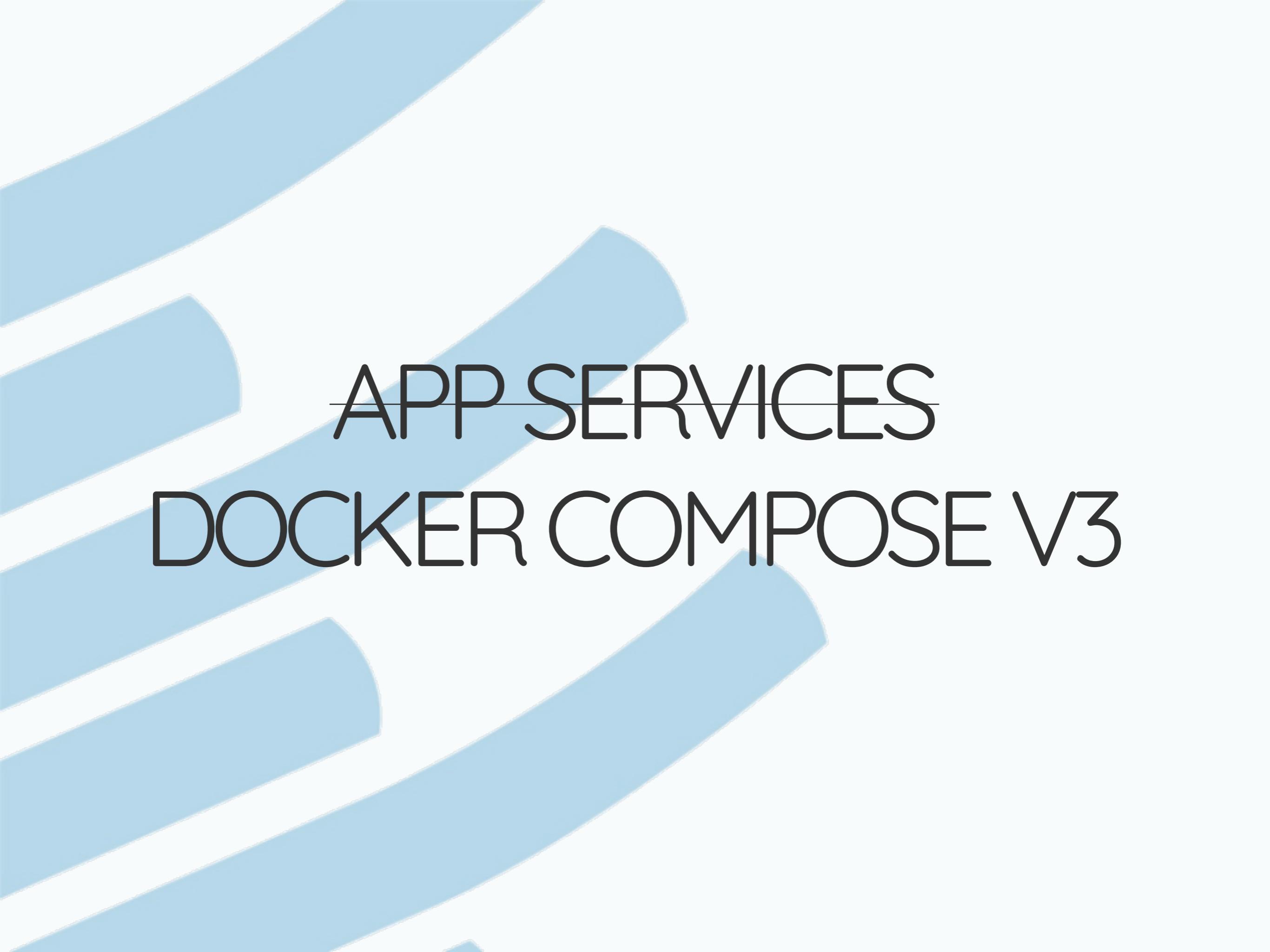


MAKFILE

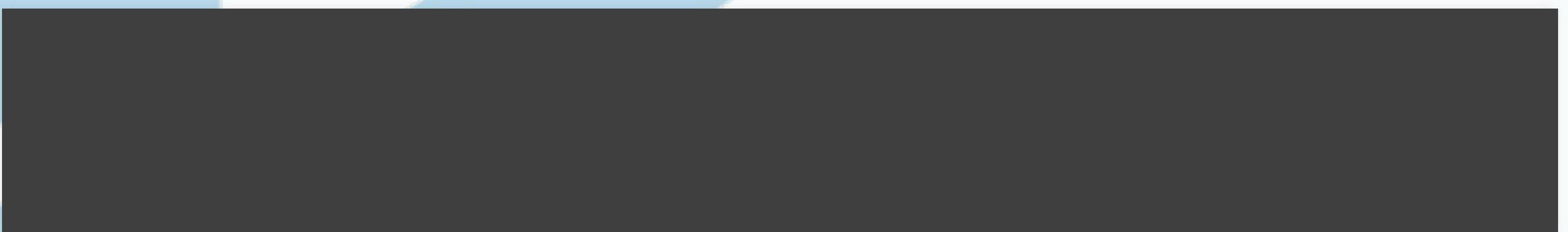
CI AAS

- Codehip
- Wercker
- Codefresh
- etc...





APP SERVICES DOCKER COMPOSE V3



ZEND SERVER

<https://github.com/5square/docker-zendserver>

HELPERS

/run.sh

```
source /shell_functions.rc
```

```
trap "remove_from_cluster; exit" SIGINT SIGTERM SIGHUP
```

WEB API

./etc/shell_functions.rc

```
WEB_API_KEY=docker
```

```
WEB_API_SECRET=$(cat /webapi/secret)
```

```
ZS_MANAGE=/usr/local/zend/bin/zs-manage
```

CONFIG IMPORT (BUILD TIME)

```
$ZS_CLIENT_A configurationImport \  
--configFile="/zs_config.zip" \  
--output-format="kv"
```

JOINING CLUSTER

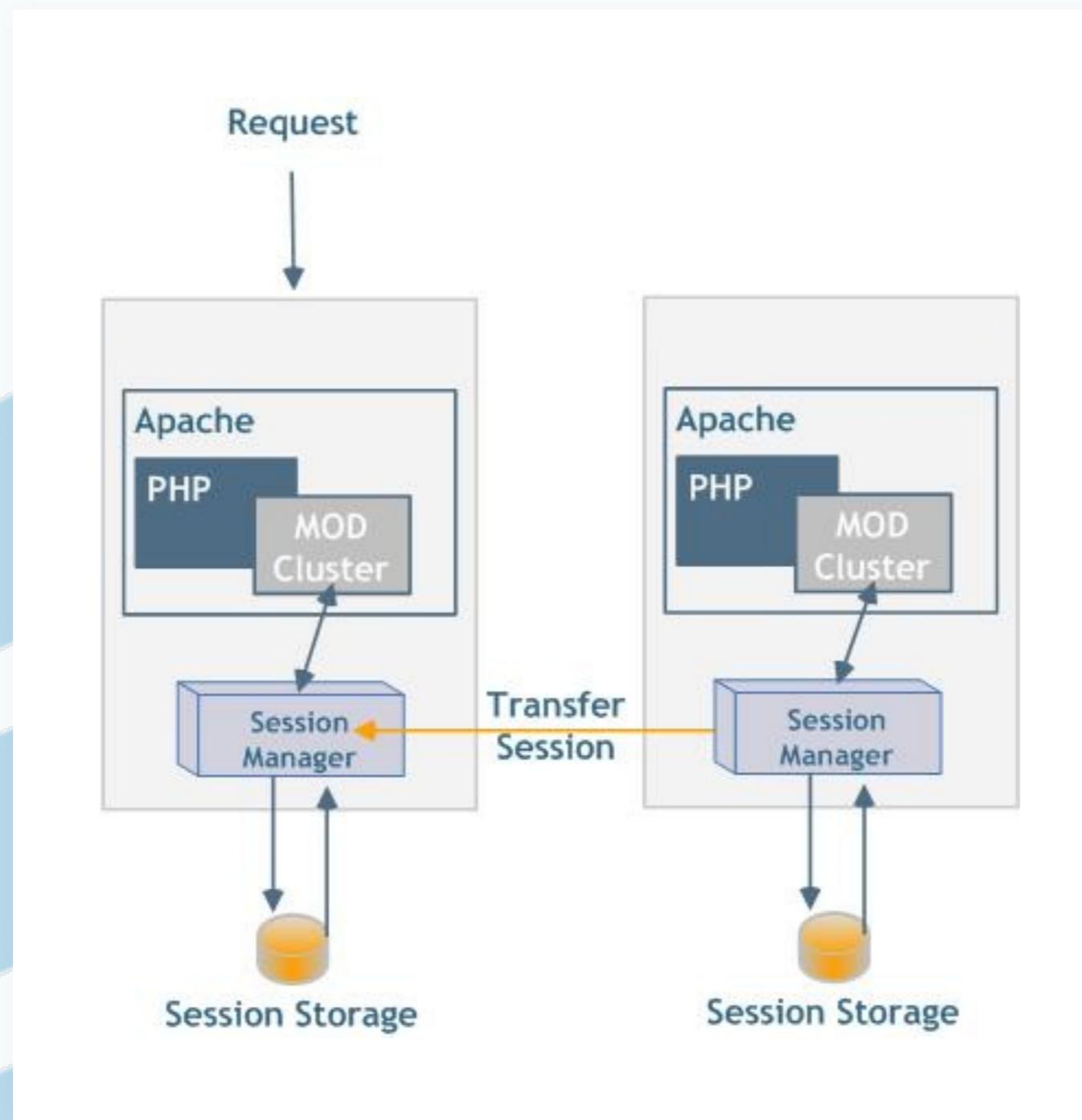
```
$ZS_MANAGE server-add-to-cluster \
-n $HOSTNAME \
-i $IP \
-o $DB_HOST:3306 -u $DB_USER -p $DB_PASS -d $DB_NAME \
-N $WEB_API_KEY -K $WEB_API_SECRET -s
```

LEAVING CLUSTER

```
$ZS_MANAGE cluster-disable-server \
$SERVER_ID -N $WEB_API_KEY -K $WEB_API_SECRET -s

$ZS_MANAGE cluster-remove-server \
$SERVER_ID -N $WEB_API_KEY -K $WEB_API_SECRET -f
```

SESSION CLUSTERING



SESSION CLUSTERING

```
$ZS_MANAGE store-directive \
-d 'session.save_handler' -v 'cluster' \
-N $WEB_API_KEY -K $WEB_API_SECRET
```

HEALTH CHECK

```
$ZS_CLIENT clusterGetServerStatus \
--servers=$(cat /usr/local/zend/etc/conf.d/ZendGlobalDir \
--zsurl="http://localhost:10081" \
--zskey=$WEB_API_KEY --zssecret=$WEB_API_SECRET \
--output-format=kv \
| grep status | grep OK || exit 1
```

The background features a series of thick, light blue diagonal stripes of varying lengths, creating a sense of motion and depth. The stripes are set against a white background.

UPDATES

ZEND SERVER DEPLOYMENT?

UPDATE

```
$ docker service update pm-zs_php --help
```

Usage: docker service update [OPTIONS] SERVICE

Update a service

Options:

- args command
- config-add config
- config-rm list
- constraint-add list
- constraint-rm list

Service command args

Add or update a config

Remove a configuration fi

Add or update a placem

Remove a constraint

UPDATE

```
$ docker service update pm-zs_php \  
--image janatzend/particulum-mobile-backend-zendserve
```



OR... UI
(PORTAINER)

The background features a series of thick, light blue diagonal stripes of varying lengths, creating a dynamic, layered effect against a white background.

BENEFITS

CONTAINERS SELF-CONTAINED

- Everything in VCS
- Running in every Docker env

DOCKER COMPOSE

Stack Deployment via YAML file

DANKESCHÖN

Contact me: jan.burkl [at] roguewave.com

Follow me: @janatzend