

Run Magento 2 on Docker Swarm





Who uses docker for development?

Anyone
Running
Magento 2
on
Kubernetes?



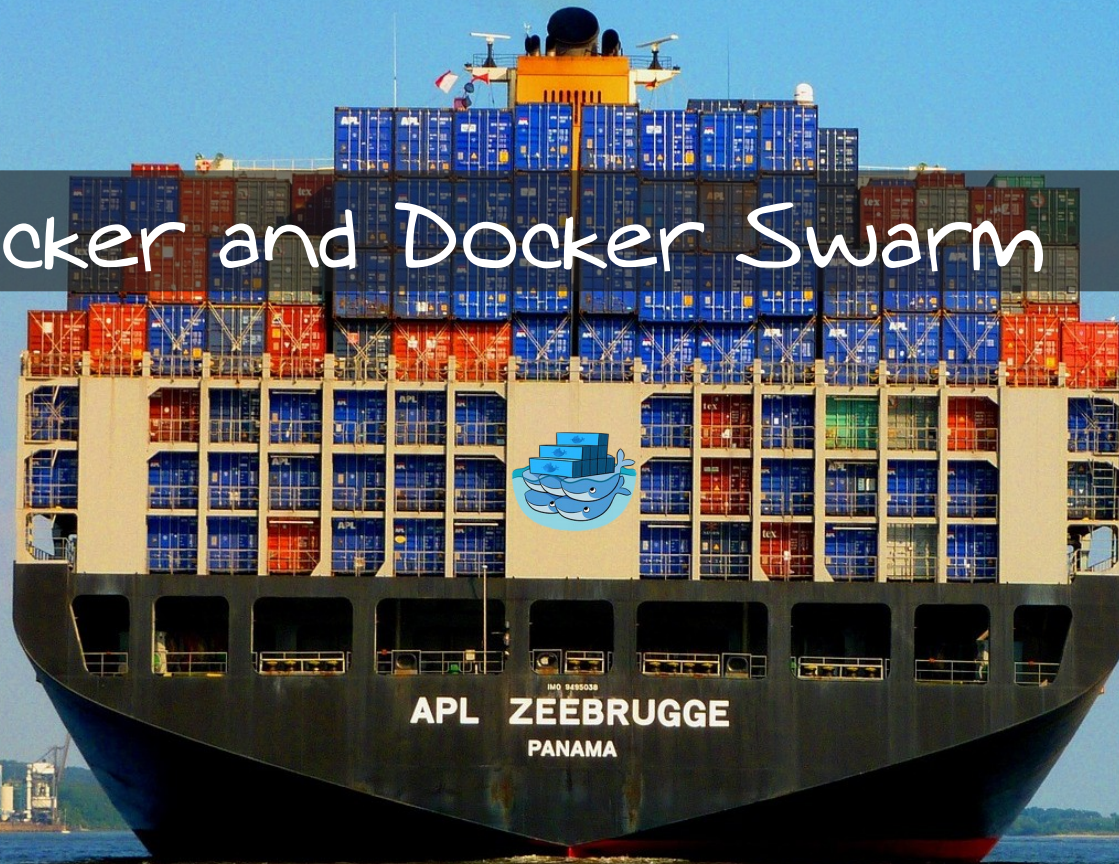


Anyone
Running
Magento 2
on
Kubernetes?



Nothing to
see here

Only Docker and Docker Swarm





Hi,

I'm Ike,

@BlackIkeEagle on Twitter

Developer / Operations
at [Studio Emma](#)



studio emma

internet expertise



Magento[®]

Silver solution partner

PIMCORE[®]

Gold solution partner

An aerial photograph of a large container port. In the background, a large cargo ship is docked at a pier, with several yellow gantry cranes positioned along the quay. The foreground and middle ground are filled with hundreds of colorful shipping containers (blue, red, green, orange, white) stacked in neat rows. The containers are of various sizes and are labeled with different shipping lines. The water is visible in the upper left corner, and the sky is clear. The overall scene depicts a busy and organized logistics hub.

First: Magento 2 development on docker

First: Magento 2 development on docker

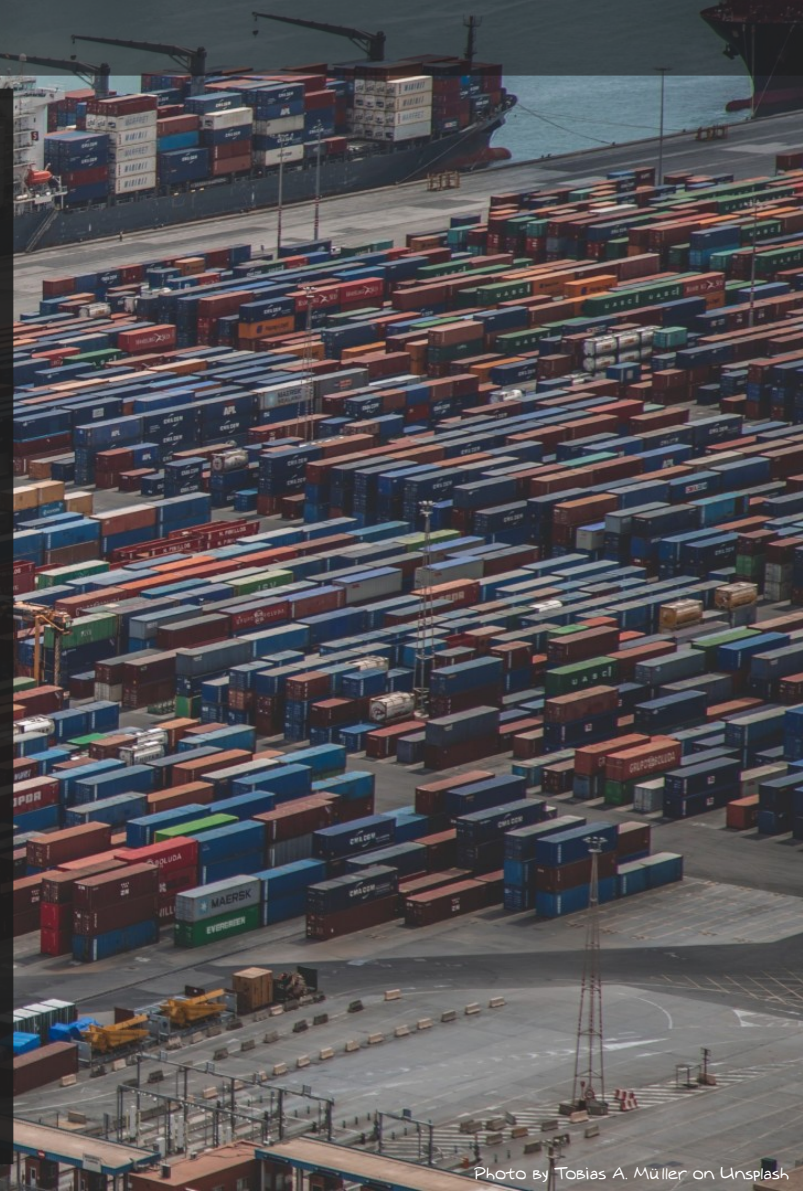
```
version: '3'
services:
  redis:
    image: redis:alpine
    environment:
      - DOMAIN_NAME=redis.${BASEHOST:-magento.docker}

  mysql:
    image: mysql:5.7
    environment:
      - MYSQL_ROOT_PASSWORD=${MYSQL_ROOT_PASSWORD:-toor}
      - DOMAIN_NAME=mysql.${BASEHOST:-magento.docker}
    command: --innodb-doublewrite=0

  mailcatcher:
    image: schickling/mailcatcher
    environment:
      - DOMAIN_NAME=mailcatcher.${BASEHOST:-magento.docker}

  application:
    image: dockerwest/php-magento${MAGENTOVERSION}:${PHPVERSION:-7.0}
    environment:
      - C_UID=${C_UID:-1000}
      - C_GID=${G_UID:-1000}
      - DEVELOPMENT=${DEVELOPMENT:-1}
    volumes:
      - ${APPLICATION:-../magento}:/phpapp

  nginx:
    image: dockerwest/nginx-magento${MAGENTOVERSION}:${NGINXVERSION:-stable}
    environment:
      - VIRTUAL_HOST=${BASEHOST:-magento.docker},${EXTRAHOSTS}
    volumes:
      - ${APPLICATION:-../magento}:/phpapp
```



First: Magento 2 development on docker

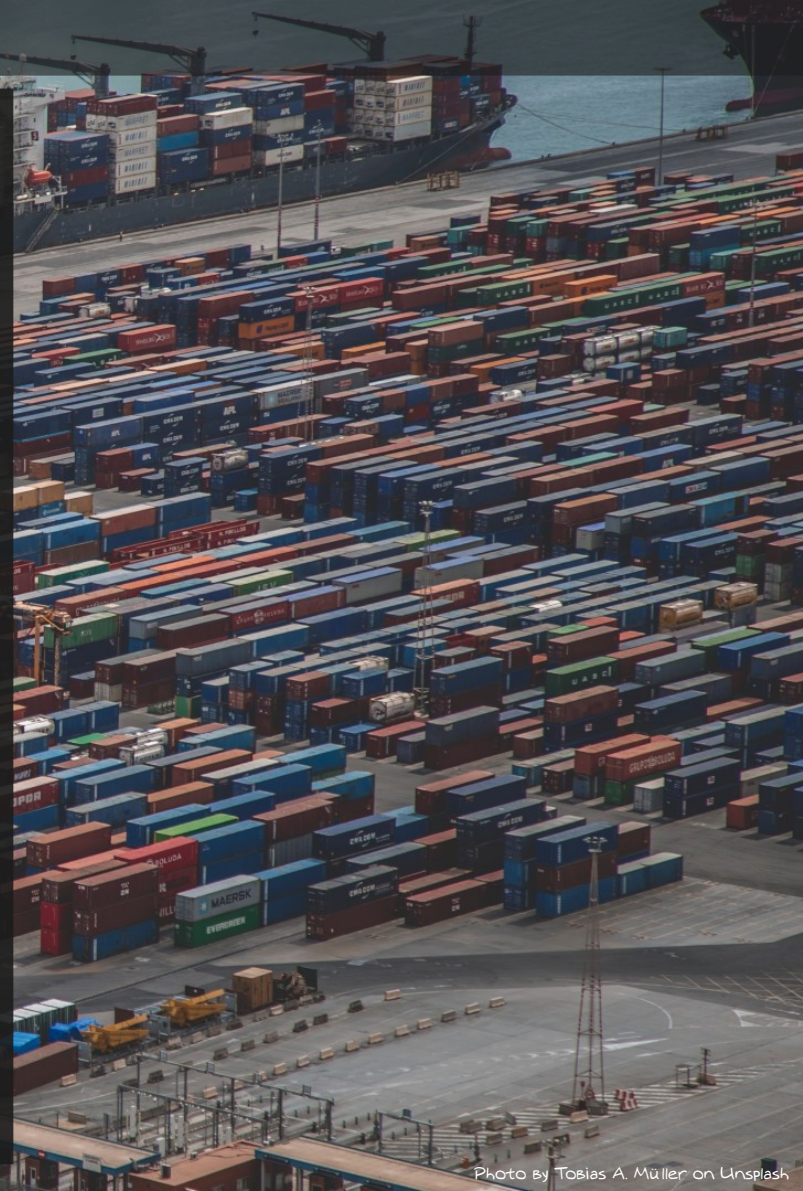
```
version: '3'
services:
  redis:
    image: redis:alpine
    environment:
      - DOMAIN_NAME=redis.${BASEHOST:-magento.docker}

  mysql:
    image: mysql:5.7
    environment:
      - MYSQL_ROOT_PASSWORD=${MYSQL_ROOT_PASSWORD:-toor}
      - DOMAIN_NAME=mysql.${BASEHOST:-magento.docker}
    command: --innodb-doublewrite=0

  mailcatcher:
    image: schickling/mailcatcher
    environment:
      - DOMAIN_NAME=mailcatcher.${BASEHOST:-magento.docker}

  application:
    image: dockerwest/php-magento${MAGENTOVERSION}:${PHPVERSION:-7.0}
    environment:
      - C_UID=${C_UID:-1000}
      - C_GID=${G_UID:-1000}
      - DEVELOPMENT=${DEVELOPMENT:-1}
    volumes:
      - ${APPLICATION:-../magento}:/phpapp

  nginx:
    image: dockerwest/nginx-magento${MAGENTOVERSION}:${NGINXVERSION:-stable}
    environment:
      - VIRTUAL_HOST=${BASEHOST:-magento.docker},${EXTRAHOSTS}
    volumes:
      - ${APPLICATION:-../magento}:/phpapp
```



First: Magento 2 development on docker

```
version: '3'
services:
  redis:
    image: redis:alpine
    environment:
      - DOMAIN_NAME=redis.${BASEHOST:-magento.docker}

  mysql:
    image: mysql:5.7
    environment:
      - MYSQL_ROOT_PASSWORD=${MYSQL_ROOT_PASSWORD:-toor}
      - DOMAIN_NAME=mysql.${BASEHOST:-magento.docker}
    command: --innodb-doublewrite=0

  mailcatcher:
    image: schickling/mailcatcher
    environment:
      - DOMAIN_NAME=mailcatcher.${BASEHOST:-magento.docker}

  application:
    image: dockerwest/php-magento${MAGENTOVERSION}:${PHPVERSION:-7.0}
    environment:
      - C_UID=${C_UID:-1000}
      - C_GID=${G_UID:-1000}
      - DEVELOPMENT=${DEVELOPMENT:-1}
    volumes:
      - ${APPLICATION:-../magento}:/phpapp

  nginx:
    image: dockerwest/nginx-magento${MAGENTOVERSION}:${NGINXVERSION:-stable}
    environment:
      - VIRTUAL_HOST=${BASEHOST:-magento.docker},${EXTRAHOSTS}
    volumes:
      - ${APPLICATION:-../magento}:/phpapp
```

```
C_UID=1000
C_GID=1000
```

```
PHPVERSION=7.0
```

```
NGINXVERSION=stable
```

```
MAGENTOVERSION=2
```

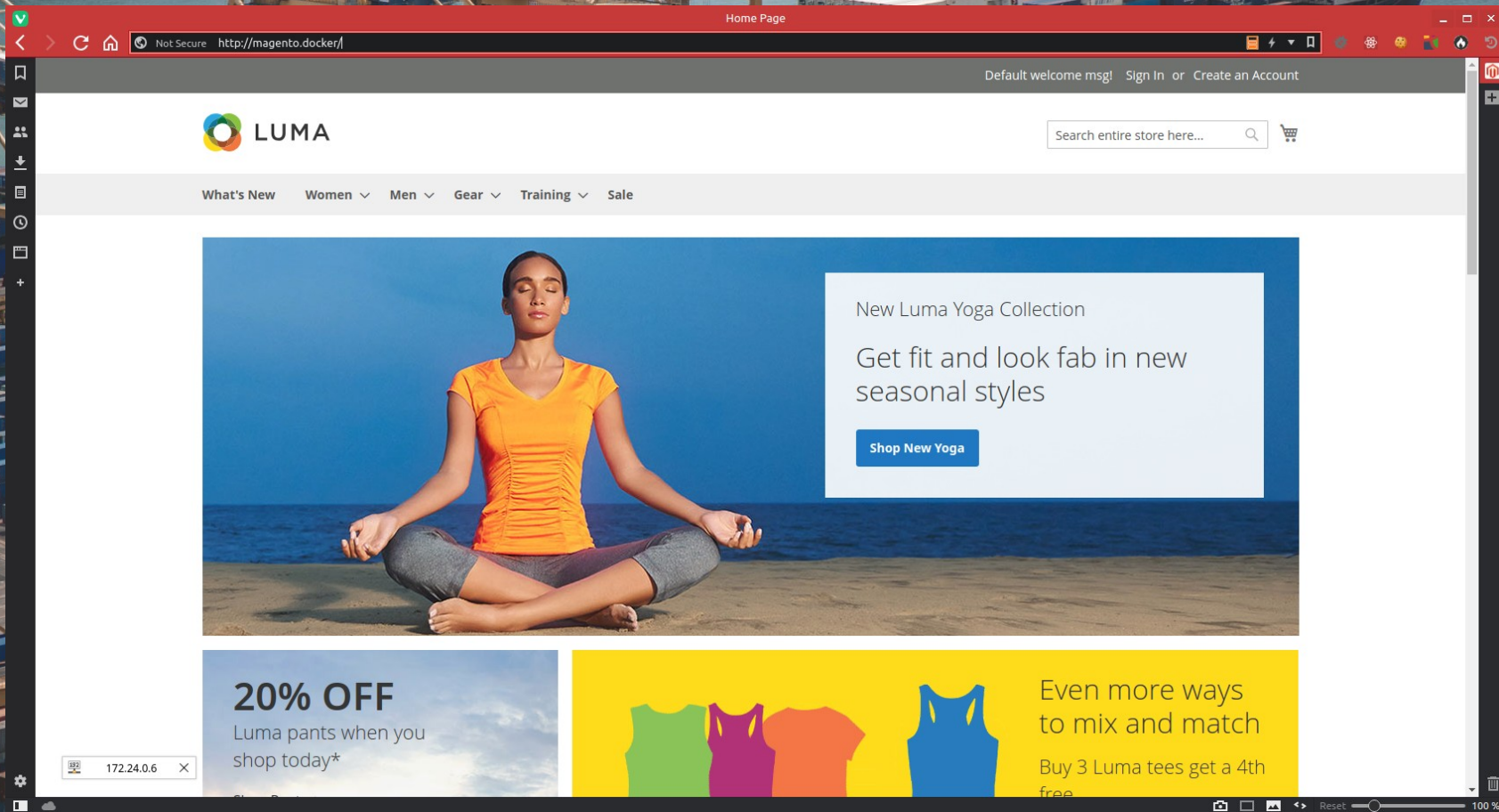
```
BASEHOST=magento.docker
EXTRAHOSTS=www.magento.docker
```

```
MYSQL_ROOT_PASSWORD=toor
```

```
APPLICATION=../magento
```

```
DEVELOPMENT=1
```

First: Magento 2 development on docker



First: Magento 2 development on docker

```
indexer:show-mode          Shows Index Mode
indexer:status             Shows status of Indexer
info
info:adminurl              Displays the Magento Admin URI
info:backups:list          Prints list of available backup files
info:currency:list         Displays the list of available currencies
info:dependencies:show-framework Shows number of dependencies on Magento framework
info:dependencies:show-modules Shows number of dependencies between modules
info:dependencies:show-modules-circular Shows number of circular dependencies between modules
info:language:list         Displays the list of available language locales
info:timezone:list         Displays the list of available timezones
maintenance
maintenance:allow-ips      Sets maintenance mode exempt IPs
maintenance:disable        Disables maintenance mode
maintenance:enable         Enables maintenance mode
maintenance:status         Displays maintenance mode status
module
module:disable             Disables specified modules
module:enable              Enables specified modules
module:status              Displays status of modules
module:uninstall           Uninstalls modules installed by composer
newrelic
newrelic:create:deploy-marker Check the deploy queue for entries and create an appropriate deploy marker.
sampledata
sampledata:deploy          Deploy sample data modules
sampledata:remove          Remove all sample data packages from composer.json
sampledata:reset           Reset all sample data modules for re-installation
setup
setup:backup               Takes backup of Magento Application code base, media and database
setup:config:set           Creates or modifies the deployment configuration
setup:cron:run             Runs cron job scheduled for setup application
setup:db-data:upgrade      Installs and upgrades data in the DB
setup:db-schema:upgrade    Installs and upgrades the DB schema
setup:db-status:upgrade    Checks if DB schema or data requires upgrade
setup:di:compile           Generates DI configuration and all missing classes that can be auto-generated
setup:install              Installs the Magento application
setup:performance:generate-fixtures Generates fixtures
setup:rollback             Rolls back Magento Application codebase, media and database
setup:static-content:deploy Deploys static view files
setup:store-config:set     Installs the store configuration. Depreciated since 2.2.0. Use config:set instead
setup:uninstall            Uninstalls the Magento application
setup:upgrade              Upgrades the Magento application, DB data, and schema
store
store:list                 Displays the list of stores
store:website:list         Displays the list of websites
theme
theme:uninstall            Uninstalls theme
varnish
varnish:vcl:generate       Generates Varnish VCL and echos it to the command line
```

```
22:01:55 tke @ archlinux-4BH8H2 ~/tmp/magento2-docker/magento
```

```
[magento@i]zsh
```

Terminal

```
nto/Luma/nl_BE/Magento_Catalog/js/product/storage/ids-storage.js HTTP/1.1" 200 834 "http://magento.docker/"
/" Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.36 Safari/537
.36 Vivaldi/2.1.1323.1" "-"
application_1 | 172.24.0.6 - 03/Oct/2018:20:02:05 +0000 "GET /static.php" 200
nginx_1 | 172.24.0.1 - - [03/Oct/2018:20:02:05 +0000] "GET /static/version1538596864/frontend/Mage
nto/Luma/nl_BE/Magento_Catalog/js/product/storage/data-storage.js HTTP/1.1" 200 1858 "http://magento.dock
er/" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.36 Safari/5
37.36 Vivaldi/2.1.1323.1" "-"
application_1 | 172.24.0.6 - 03/Oct/2018:20:02:05 +0000 "GET /static.php" 200
nginx_1 | 172.24.0.1 - - [03/Oct/2018:20:02:05 +0000] "GET /static/version1538596864/frontend/Mage
nto/Luma/nl_BE/Magento_Catalog/js/product/storage/ids-storage-compare.js HTTP/1.1" 200 709 "http://magento
o.docker/" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.36 Sa
fari/537.36 Vivaldi/2.1.1323.1" "-"
application_1 | 172.24.0.6 - 03/Oct/2018:20:02:05 +0000 "GET /static.php" 200
nginx_1 | 172.24.0.1 - - [03/Oct/2018:20:02:05 +0000] "GET /static/version1538596864/frontend/Mage
nto/Luma/nl_BE/Magento_UI/js/form/adapters/buttons.js HTTP/1.1" 200 217 "http://magento.docker/" "Mozilla/
5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.36 Safari/537.36 Vivaldi/
2.1.1323.1" "-"
application_1 | 172.24.0.6 - 03/Oct/2018:20:02:05 +0000 "GET /static.php" 200
nginx_1 | 172.24.0.1 - - [03/Oct/2018:20:02:05 +0000] "GET /static/version1538596864/frontend/Mage
nto/Luma/nl_BE/Magento_UI/templates/collection.html HTTP/1.1" 200 192 "http://magento.docker/" "Mozilla/5
.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.36 Safari/537.36 Vivaldi/2
.1.1323.1" "-"
application_1 | 172.24.0.6 - 03/Oct/2018:20:02:05 +0000 "GET /static.php" 200
nginx_1 | 172.24.0.1 - - [03/Oct/2018:20:02:06 +0000] "GET /static/version1538596864/frontend/Mage
nto/Luma/nl_BE/Magento_Captcha/js/action/refresh.js HTTP/1.1" 200 303 "http://magento.docker/" "Mozilla/5
.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.36 Safari/537.36 Vivaldi/2
.1.1323.1" "-"
application_1 | 172.24.0.6 - 03/Oct/2018:20:02:05 +0000 "GET /index.php" 200
nginx_1 | 172.24.0.1 - - [03/Oct/2018:20:02:06 +0000] "GET /customer/section/load/?sections=cart&u
pdate_section_id=false&=1538596922558 HTTP/1.1" 200 226 "http://magento.docker/" "Mozilla/5.0 (X11; Linu
x x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.36 Safari/537.36 Vivaldi/2.1.1323.1" "-"
application_1 | 172.24.0.6 - 03/Oct/2018:20:02:05 +0000 "GET /static.php" 200
nginx_1 | 172.24.0.1 - - [03/Oct/2018:20:02:06 +0000] "GET /static/version1538596864/frontend/Mage
nto/Luma/nl_BE/Magento_Catalog/js/product/query-builder.js HTTP/1.1" 200 388 "http://magento.docker/" "Mo
zilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.36 Safari/537.36 Vi
valdi/2.1.1323.1" "-"
application_1 | 172.24.0.6 - 03/Oct/2018:20:02:05 +0000 "GET /static.php" 200
nginx_1 | 172.24.0.1 - - [03/Oct/2018:20:02:06 +0000] "GET /static/version1538596864/frontend/Mage
nto/Luma/nl_BE/Magento_Customer/template/authentication-popup.html HTTP/1.1" 200 1128 "http://magento.doc
ker/" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.36 Safari/
537.36 Vivaldi/2.1.1323.1" "-"
application_1 | 172.24.0.6 - 03/Oct/2018:20:02:06 +0000 "GET /static.php" 200
nginx_1 | 172.24.0.1 - - [03/Oct/2018:20:02:06 +0000] "GET /static/version1538596864/frontend/Mage
nto/Luma/nl_BE/Magento_UI/template/messages.html HTTP/1.1" 200 336 "http://magento.docker/" "Mozilla/5.0
(X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.36 Safari/537.36 Vivaldi/2.1.
1323.1" "-"
application_1 | 172.24.0.1 - - [03/Oct/2018:20:02:06 +0000] "GET /static/version1538596864/frontend/Mage
nto/Luma/nl_BE/Magento_Captcha/template/checkout/captcha.html HTTP/1.1" 200 672 "http://magento.docker/"
"Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.36 Safari/537.36
Vivaldi/2.1.1323.1" "-"
application_1 | 127.0.0.1 - 03/Oct/2018:20:02:08 +0000 "GET /ping" 200
```

```
archlinux-4BH8H2 2018-10-03 22:02:07
```



Second: Docker Swarm

Second: Docker Swarm

Node

Node

Node

Node

Second: Docker Swarm

Node

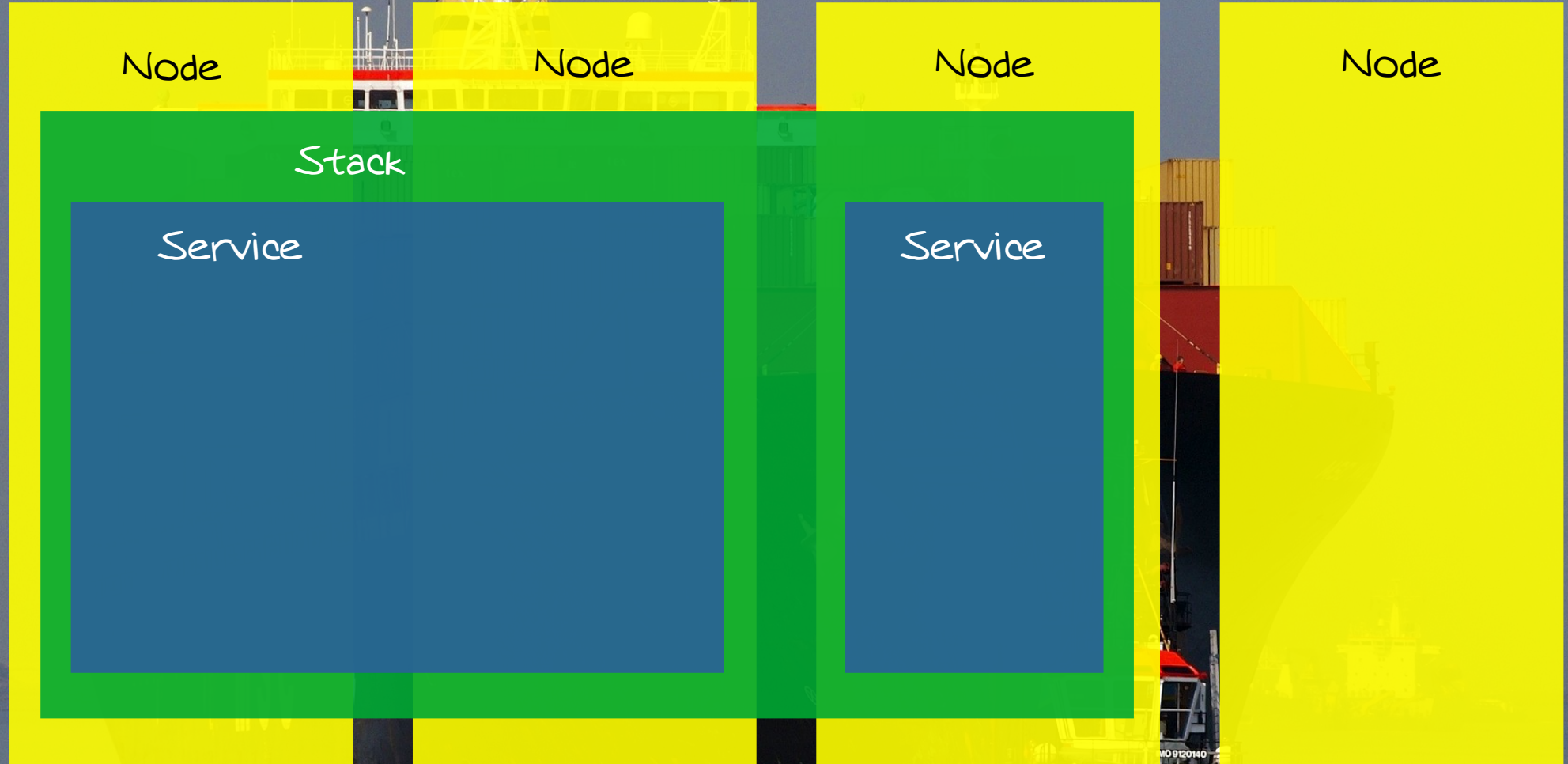
Node

Node

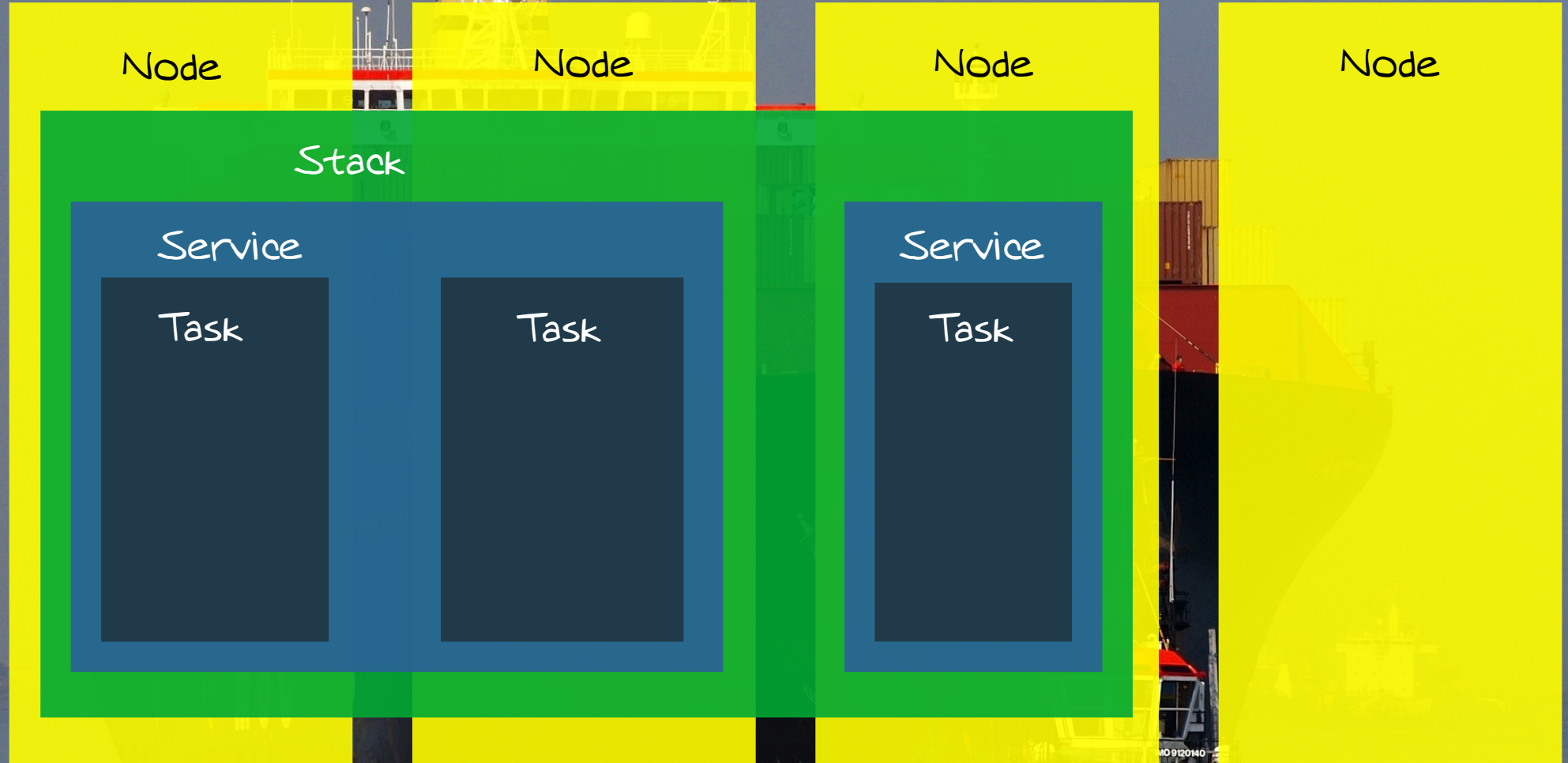
Node

Stack

Second: Docker Swarm



Second: Docker Swarm



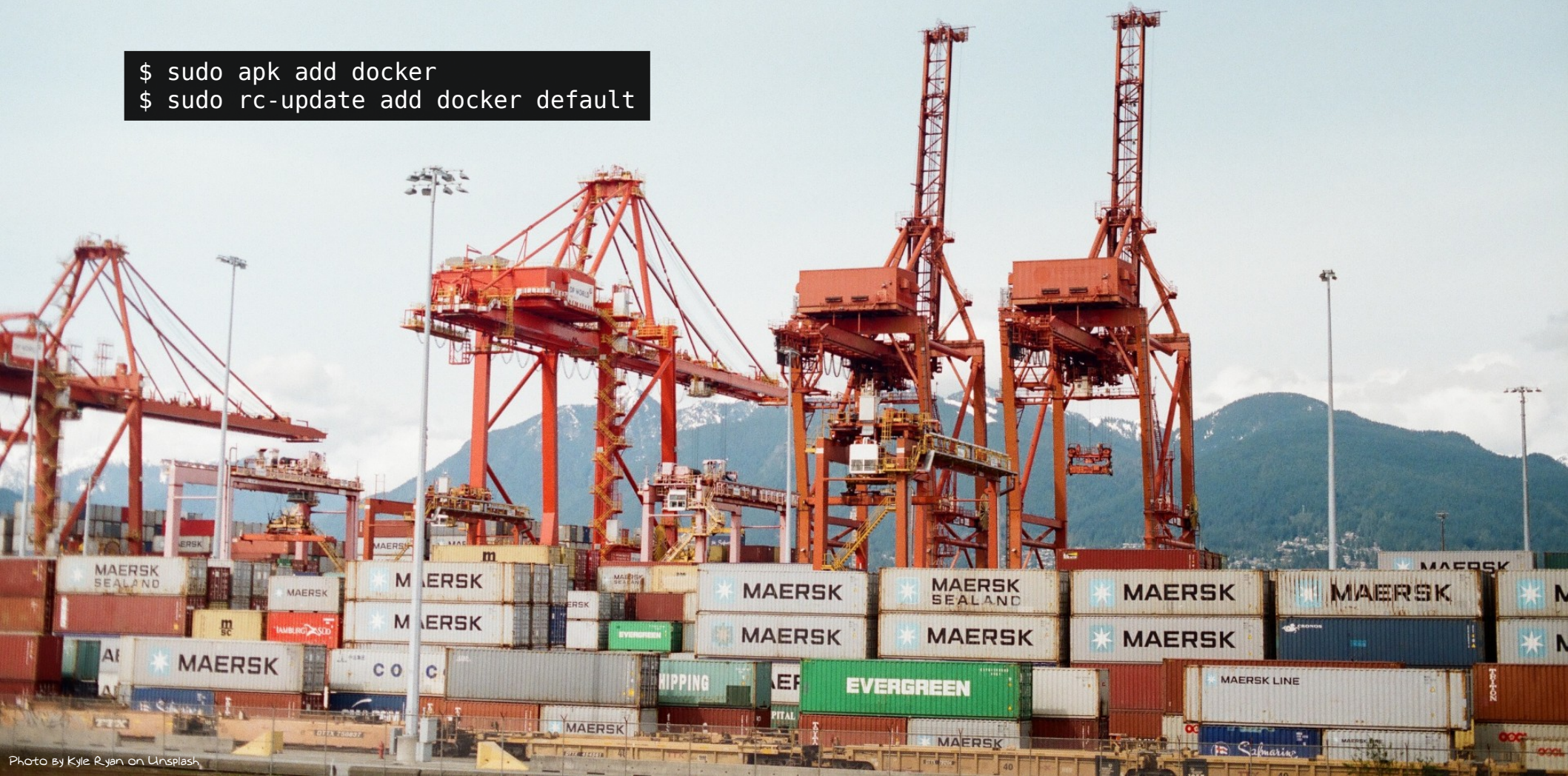
Third: Docker Swarm setup



Third: Docker Swarm setup

Docker

```
$ sudo apt add docker  
$ sudo rc-update add docker default
```



Third: Docker Swarm setup

Docker

```
$ sudo apt add docker  
$ sudo rc-update add docker default
```

```
$ sudo apt-get install docker-ce  
$ sudo systemctl enable docker.service
```

Third: Docker Swarm setup

Docker

```
$ sudo apk add docker  
$ sudo rc-update add docker default
```

```
$ sudo apt-get install docker-ce  
$ sudo systemctl enable docker.service
```

```
$ sudo yum install docker-ce  
$ sudo systemctl enable docker.service
```

Third: Docker Swarm setup

Secure communication

```
[ -d client-cert ] && rm -r client-cert
[ -d server-cert ] && rm -r server-cert

mkdir -p client-cert
docker run --user=$(id -u) --rm -v $(pwd)/client-cert:/certs paulczar/omgwtfssl

mkdir -p server-cert
cp client-cert/ca.pem server-cert/
docker run --user=$(id -u) --rm -v $(pwd)/server-cert:/server \
-v $(pwd)/client-cert:/certs \
-e SSL_IP=127.0.0.1,192.168.250.31,192.168.250.32 \
-e SSL_DNS=docker.local -e SSL_KEY=/server/key.pem \
-e SSL_CERT=/server/cert.pem paulczar/omgwtfssl

docker_server_args= \
"-H unix:///var/run/docker.sock -H tcp://0.0.0.0:2375" \
"--tlsverify --tlscacert=/etc/docker/ssl/ca.pem " \
"--tlscert=/etc/docker/ssl/cert.pem" \
"--tlskey=/etc/docker/ssl/key.pem"
```



Third: Docker Swarm setup

Secure communication

```
[ -d client-cert ] && rm -r client-cert
[ -d server-cert ] && rm -r server-cert

mkdir -p client-cert
docker run --user=$(id -u) --rm -v $(pwd)/client-cert:/certs paulczar/omgwtfssl

mkdir -p server-cert
cp client-cert/ca.pem server-cert/
docker run --user=$(id -u) --rm -v $(pwd)/server-cert:/server \
-v $(pwd)/client-cert:/certs \
-e SSL_IP=127.0.0.1,192.168.250.31,192.168.250.32 \
-e SSL_DNS=docker.local -e SSL_KEY=/server/key.pem \
-e SSL_CERT=/server/cert.pem paulczar/omgwtfssl

docker server_args= \
-H unix:///var/run/docker.sock -H tcp://0.0.0.0:2375" \
"--tlsverify --tlscacert=/etc/docker/ssl/ca.pem " \
"--tlscert=/etc/docker/ssl/cert.pem" \
"--tlskey=/etc/docker/ssl/key.pem"
```

```
export DOCKER_HOST=tcp://192.168.250.31:2375
export DOCKER_CERT_PATH=$(pwd)/client-cert
export DOCKER_TLS_VERIFY=1
```

Third: Docker Swarm setup

Initialize Swarm

```
$ sudo docker swarm init --advertise-addr=192.168.250.31  
$ managertoken=$(docker swarm join-token -q manager)  
$ sudo docker swarm join --token $managertoken 192.168.250.31  
$ workertoken=$(docker swarm join-token -q worker)  
$ sudo docker swarm join --token $workertoken 192.168.250.31
```

Demo



Third: Docker Swarm setup Extras Persistent Storage

Nfs, ceph, gluster, ...

Third: Docker Swarm setup Extras Visualization

swarm visualizer

filter containers

manager-1 manager	manager-2 manager	node-1 worker	node-2 worker
2.934G RAM x86_64/linux	2.934G RAM x86_64/linux	1.460G RAM x86_64/linux	1.460G RAM x86_64/linux
magento2_application image: swarmprom-magento2-demo-phy tag: 0.0.2@sha256:b6a5af30bbcb0c updated: 5/10 9:16 7af345c95407618545e1947cbf13de state: running mon_prometheus image: swarmprom-prometheus-lat tag: latest@sha256:5928441c29403 cmd: -config.file=/etc/prometheus/... updated: 5/10 9:16 6951701c08f39a095795f38312ac87 state: running magento2_mysql image: mysql:5.7@sha256:1d8471c tag: 5.7.30@sha256:1d8471c76292ec cmd: --innodb-doublewrite=0 state: running	viz_visualizer image: visualizer:latest@sha256:1f1e tag: latest@sha256:1f1e4f29c43641 updated: 5/10 9:22 52190cb308bec0c511d58816b36be5 state: running mon_dockerd-exporter image: caddy:latest@sha256:44541f4cb6694 tag: latest@sha256:44541f4cb6694 updated: 5/10 9:16 bb9ad078ec1f0b580461db389dfe1a state: running magento2_application image: swarm-magento2-demo-phy tag: 0.0.2@sha256:b6a5af30bbcb0c updated: 5/10 9:16 e281cbe7e60eac2087d6a63c12539 state: running	mon_node-exporter image: swarmprom-node-exporter tag: latest@sha256:ee424a4176023 cmd: --collector.textfile.directory=/... updated: 5/10 9:16 d68c0b32766d272822cedb661d29 state: running portainer_agent image: agent:latest@sha256:468f5b tag: latest@sha256:468f5b8ed0a81 updated: 5/10 9:16 4d545e514abb25ec9b59673bef4c state: running magento2_redis image: redis:alpine@sha256:2953c5 tag: alpine@sha256:2953c53708ba updated: 5/10 9:16 state: running	magento2_nginx image: swarm-magento2-demo-nginx tag: 0.0.2@sha256:126d962d5558b updated: 5/10 9:16 ba89683c573d185907f0e2d77818 state: running mon_node-exporter image: swarmprom-node-exporter tag: latest@sha256:ee424a4176023 cmd: --collector.textfile.directory=/... updated: 5/10 9:16 84c4724a2db8069058b65e0e7f3 state: running portainer_agent image: agent:latest@sha256:468f5b tag: latest@sha256:468f5b8ed0a81 updated: 5/10 9:16 state: running

portainer

portainer.io

Dashboard
Endpoint summary

Portainer support admin
my account log out

Cluster information

Nodes in the cluster 4
[Go to cluster visualizer](#)

4 Stacks

17 Services

41 Containers
30 running
11 stopped

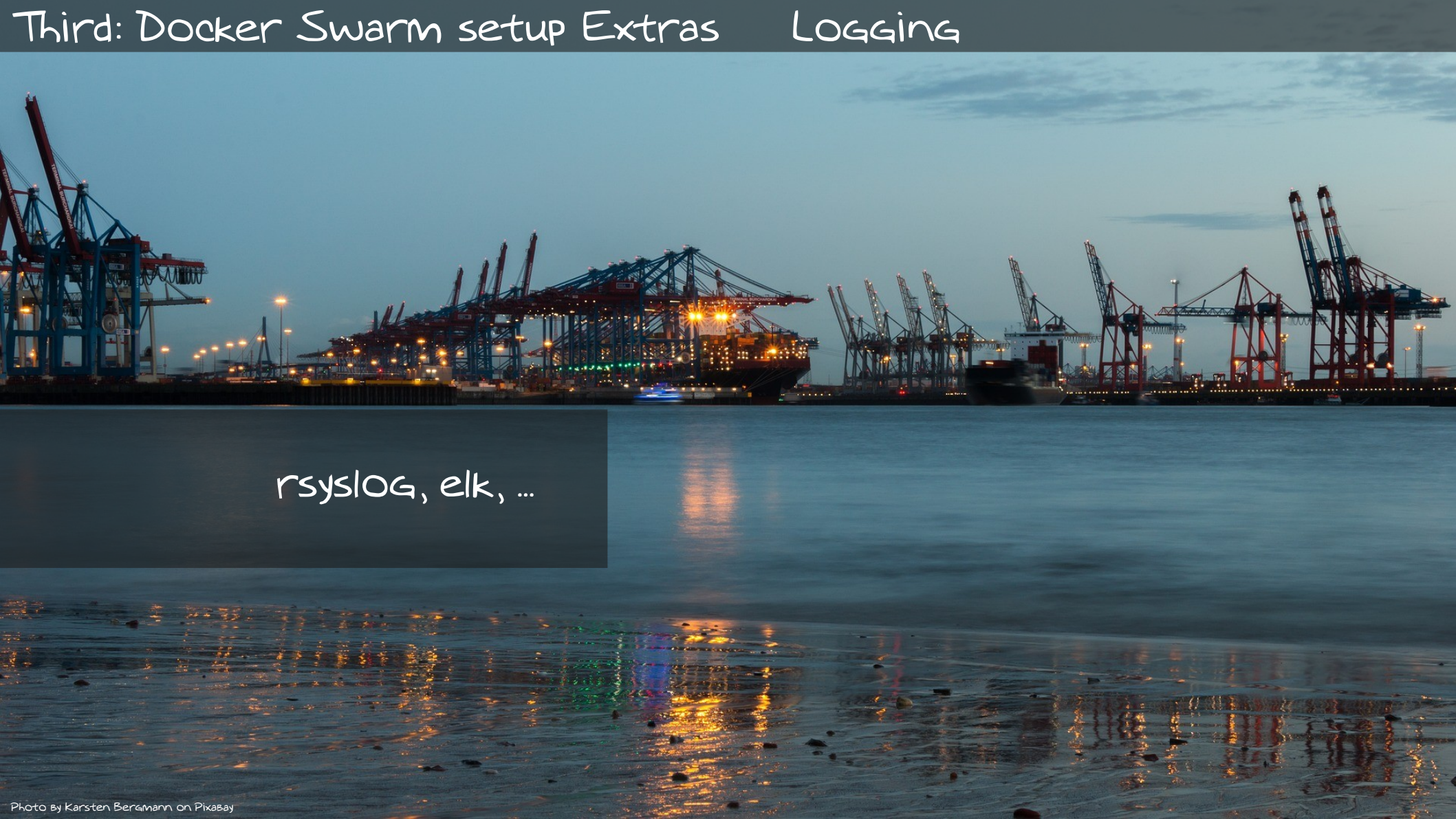
30 Images
5 GB

29 Volumes

35 Networks

192.168.250.31

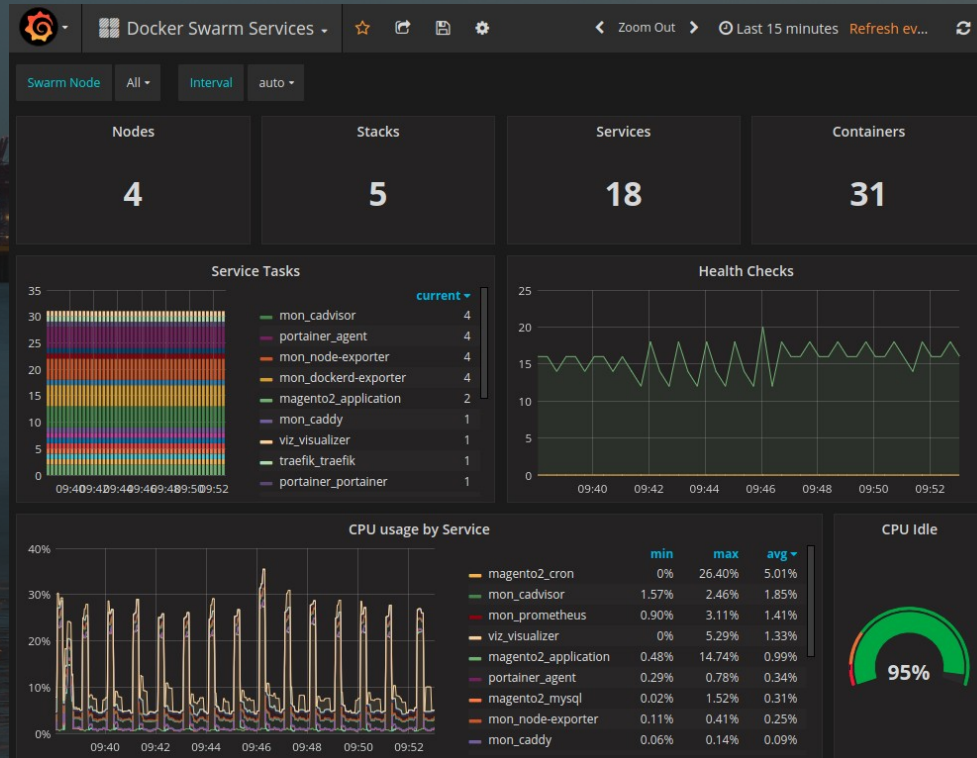
Third: Docker Swarm setup Extras Logging



rsyslog, elk, ...

Third: Docker Swarm setup Extras Monitoring

ELK, zabbix, prometheus + grafana, ...



Third: Docker Swarm setup Extras Proxy (Handle multiple apps)



haproxy, nginx, traefik, ...

Demo



Fourth: Magento 2 on Swarm



- webserver (nginx)
- php (php-fpm)
- cache (redis)
- dB (mysql)
- cron



```
FROM dockerwest/nginx-magento2:mainline
```

```
COPY ./pub /phpapp/pub
```

```
RUN rm -rf /phpapp/pub/media/* && chown -R nginx:nginx /phpapp/pub
```



```
FROM dockerwest/php-magento2:7.1
```

```
COPY ./entrypoint.sh /
```

```
COPY ./go-cron /usr/local/bin
```

```
COPY ./ /phpapp/
```

```
RUN rm -rf /phpapp/var/* && rm -rf /phpapp/pub/media/* \  
&& rm -rf /phpapp/update && rm -rf /phpapp/phpserver && rm -rf /phpapp/dev \  
&& rm -f /phpapp/app/etc/env.php \  
&& chown -R www-data:www-data /phpapp
```

```
ENTRYPOINT ["/entrypoint.sh"]
```

```
CMD ["php-fpm7.1"]
```



```
FROM dockerwest/php-magento2:7.1
```

```
COPY ./entrypoint.sh /
```

```
COPY ./go-cron /usr/local/bin
```

```
COPY ./ /phpapp/
```

```
RUN rm -rf /phpapp/var/* && rm -rf /phpapp/pub/media/* \  
&& rm -rf /phpapp/update && rm -rf /phpapp/phpserver && rm -rf /phpapp/dev \  
&& rm -f /phpapp/app/etc/env.php \  
&& chown -R www-data:www-data /phpapp
```

```
ENTRYPOINT ["/entrypoint.sh"]
```

```
CMD ["php-fpm7.1"]
```

```
#!/bin/sh  
set -e
```

```
if [ "${CRON}" ]; then  
    exec /command.sh go-cron -s "${CRON}" -- "$@"  
else  
    exec /command.sh "$@"  
fi
```

- initial media
- prepared dB



- create volumes
- set proper permissions
- set initial media
- import db
- set proper configuration



```
version: '3.3'

networks:
  default:
    ipam:
      config:
        - subnet: 10.110.3.0/24
traefik_webproxy:
  external: true
```



```
version: '3.3'

networks:
  default:
    ipam:
      config:
        - subnet: 10.110.3.0/24
traefik_webproxy:
  external: true
```

```
volumes:
  redis:
    driver: local
    driver_opts:
      type: nfs
      o: addr=192.168.250.21,rw
      device: ":/srv/storage/magento2/redis"
  mysql:
    driver: local
    driver_opts:
      type: nfs
      o: addr=192.168.250.21,rw
      device: ":/srv/storage/magento2/mysql"
  magento_var:
    driver: local
    driver_opts:
      type: nfs
      o: addr=192.168.250.21,rw
      device: ":/srv/storage/magento2/var"
  magento_media:
    driver: local
    driver_opts:
      type: nfs
      o: addr=192.168.250.21,rw
      device: ":/srv/storage/magento2/media"
```

```
version: '3.3'

networks:
  default:
    ipam:
      config:
        - subnet: 10.110.3.0/24
traefik_webproxy:
  external: true
```

```
configs:
  magento_config:
    file: ./env.php
```

```
volumes:
  redis:
    driver: local
    driver_opts:
      type: nfs
      o: addr=192.168.250.21,rw
      device: ":/srv/storage/magento2/redis"
  mysql:
    driver: local
    driver_opts:
      type: nfs
      o: addr=192.168.250.21,rw
      device: ":/srv/storage/magento2/mysql"
  magento_var:
    driver: local
    driver_opts:
      type: nfs
      o: addr=192.168.250.21,rw
      device: ":/srv/storage/magento2/var"
  magento_media:
    driver: local
    driver_opts:
      type: nfs
      o: addr=192.168.250.21,rw
      device: ":/srv/storage/magento2/media"
```

Fourth: Magento 2 on Swarm

Up and running

```
services:
  redis:
    image: redis:alpine
    volumes:
      - type: volume
        source: redis
        target: /data
        volume:
          nocopy: true
    logging:
      driver: syslog
      options:
        tag: magento-redis
        syslog-address: "tcp://192.168.250.21:514"
```



```
services:
  redis:
    image: redis:alpine
    volumes:
      - type: volume
        source: redis
        target: /data
        volume:
          nocopy: true
    logging:
      driver: syslog
      options:
        tag: magento-redis
        syslog-address: "tcp://192.168.250.21:514"
```

```
mysql:
  image: mysql:5.7
  environment:
    - MYSQL_ROOT_PASSWORD=toor
  command: --innodb-doublewrite=0
  volumes:
    - type: volume
      source: mysql
      target: /var/lib/mysql
      volume:
        nocopy: true
  logging:
    driver: syslog
    options:
      tag: magento-mysql
      syslog-address: "tcp://192.168.250.21:514"
  deploy:
    placement:
      constraints: [node.role == manager]
```

```
application:
  image: blackikeagle/swarm-magento2-demo-php:0.0.2
  environment:
    - DEVELOPMENT=noprofile # we want to use mailcatcher in the demo
  volumes:
    - type: volume
      source: magento_var
      target: /phpapp/var
      volume:
        nocopy: true
    - type: volume
      source: magento_media
      target: /phpapp/pub/media
      volume:
        nocopy: true
  configs:
    - source: magento_config
      target: /phpapp/app/etc/env.php
  logging:
    driver: syslog
    options:
      tag: magento-application
      syslog-address: "tcp://192.168.250.21:514"
  deploy:
    replicas: 2
```

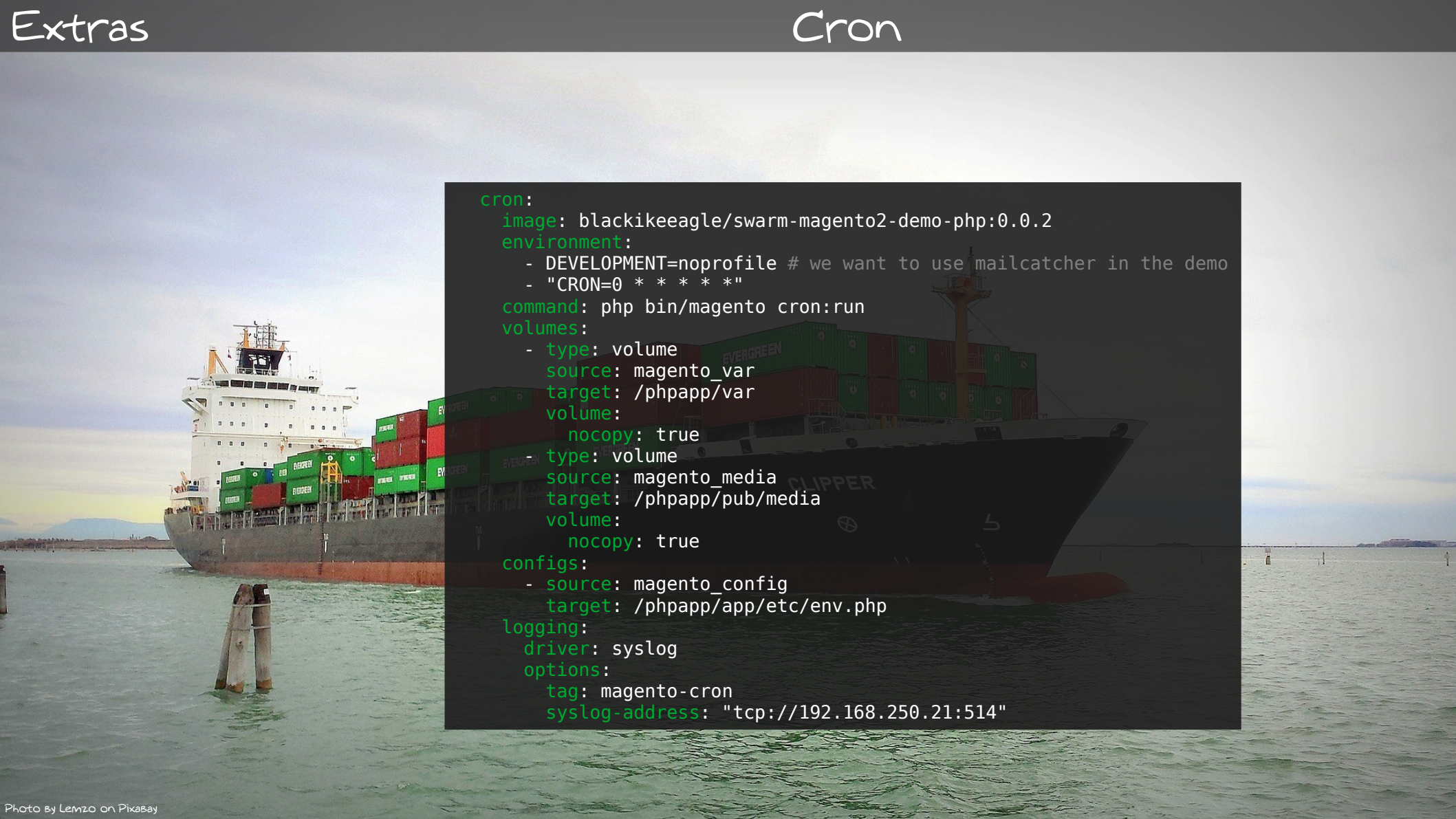
```
nginx:
  image: blackikееagle/swarm-magento2-demo-nginx:0.0.2
  volumes:
    - type: volume
      source: magento_media
      target: /phpapp/pub/media
      volume:
        nocopy: true
  networks:
    - default
    - traefik_webproxy
  logging:
    driver: syslog
    options:
      tag: magento-nginx
      syslog-address: "tcp://192.168.250.21:514"
  deploy:
    labels:
      # tell traefik to route to the container
      - "traefik.enable=true"
      # a name for this application in traefik, must be unique
      - "traefik.backend=magento2"
      # hostnames to route to this container (and retrieve ssl certs for)
      - "traefik.frontend.rule=Host:magento2-demo.test"
      # network this container is attached to (global name)
      - "traefik.docker.network=traefik_webproxy"
      # port this container listens on
      - "traefik.port=80"
```

Demo



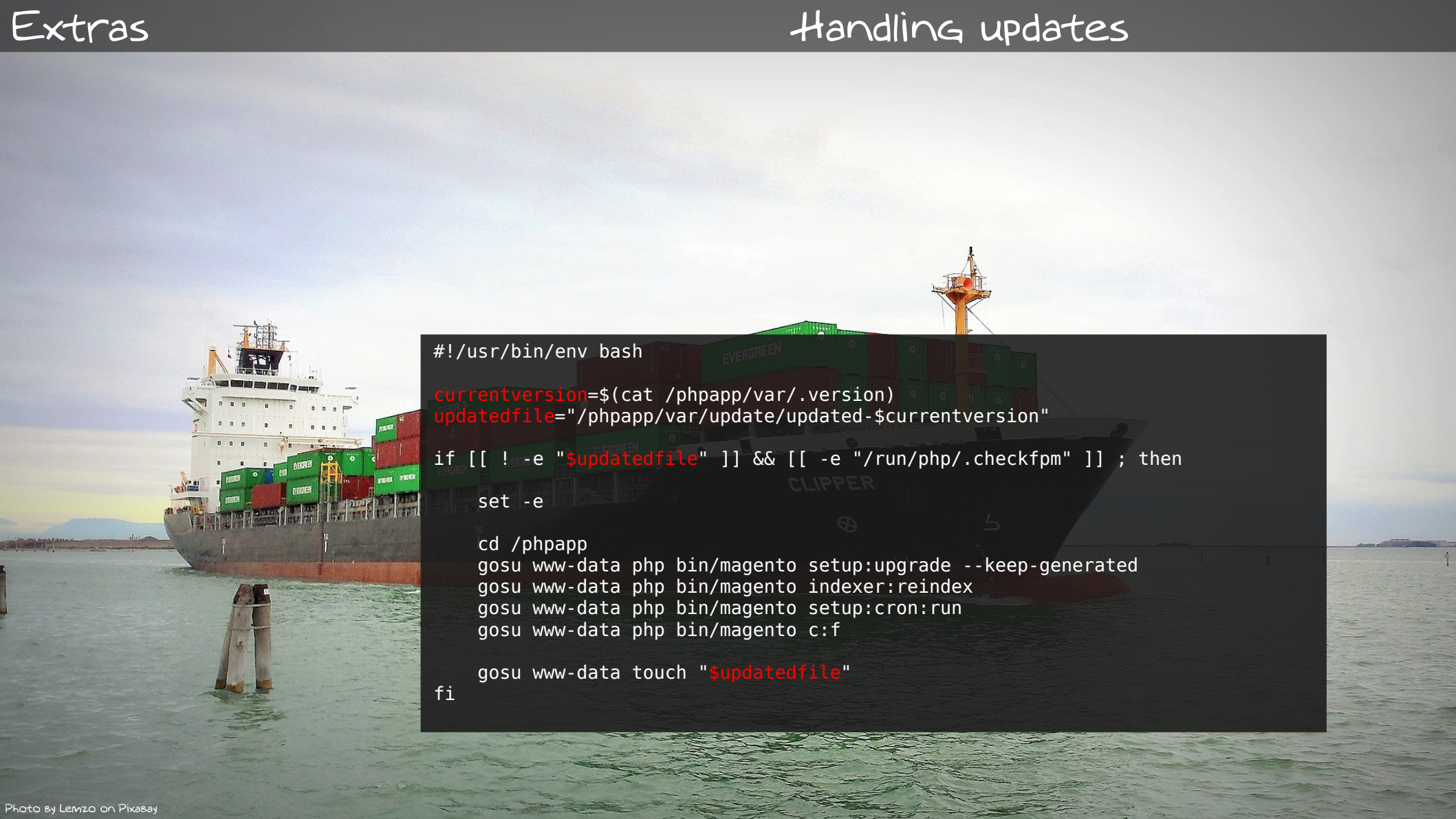
Extras





```
cron:
  image: blackikeeagle/swarm-magento2-demo-php:0.0.2
  environment:
    - DEVELOPMENT=noprofile # we want to use mailcatcher in the demo
    - "CRON=0 * * * * *"
  command: php bin/magento cron:run
  volumes:
    - type: volume
      source: magento_var
      target: /phpapp/var
      volume:
        nocopy: true
    - type: volume
      source: magento_media
      target: /phpapp/pub/media
      volume:
        nocopy: true
  configs:
    - source: magento_config
      target: /phpapp/app/etc/env.php
  logging:
    driver: syslog
    options:
      tag: magento-cron
      syslog-address: "tcp://192.168.250.21:514"
```

```
application:
  image: blackikееagle/swarm-magento2-demo-php:0.0.2
  volumes:
    - type: volume
      source: magento_var
      target: /phpapp/var
      volume:
        nocopy: true
    - type: volume
      source: magento_media
      target: /phpapp/pub/media
      volume:
        nocopy: true
  configs:
    - source: magento_config
      target: /phpapp/app/etc/env.php
  logging:
    driver: syslog
    options:
      tag: magento-application
      syslog-address: "tcp://192.168.250.21:514"
  deploy:
    replicas: 2
    update_config:
      parallelism: 1 # update per 1 running containers
      delay: 10s # wait for 10s healthy status before continuing
```



```
#!/usr/bin/env bash

currentversion=$(cat /phpapp/var/.version)
updatedfile="/phpapp/var/update/updated-$currentversion"

if [[ ! -e "$updatedfile" ]] && [[ -e "/run/php/.checkfpm" ]] ; then
    set -e

    cd /phpapp
    gosu www-data php bin/magento setup:upgrade --keep-generated
    gosu www-data php bin/magento indexer:reindex
    gosu www-data php bin/magento setup:cron:run
    gosu www-data php bin/magento c:f

    gosu www-data touch "$updatedfile"
fi
```

Extras

Mail



msmtp, mhsendmail, ...

Pro

- Easy to setup
- docker-compose → swarm is a small step
- Great management tools available
- very easy to maintain

Con

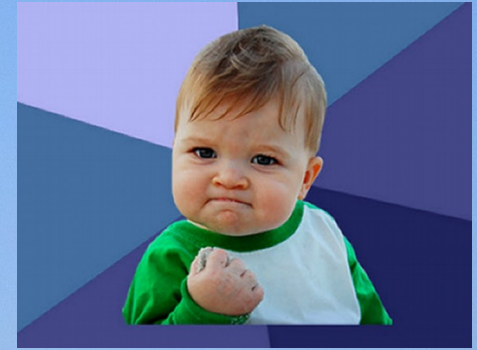
- Not all Batteries included
- no "Cron" functionality Built in
- no "One-off" functionality Built in
- no auto re-Balance



It's not because you can run mysql containerized, you should do that
Keep track of your storage, you will end up with a full disk eventually
Swarm does not have autoscaling, (out of the box)
Magento logging should be centralized in ELK



We can run Magento 2 on Docker Swarm



We can run Magento 2 on Docker Swarm



Questions?

I'm Ike, Developer / Operations
at Studio Emma



@BlackIkeEagle



PHP-WVL



DockerWest



Parchive



BlackIkeEagle



Archlinux TU



Vdebug



herecura.eu



Thanks

<https://joind.in/talk/balaf>

I'm Ike, Developer / Operations
at Studio Emma



@BlackIkeEagle



PHP-WVL



DockerWest



Parchive



BlackIkeEagle



Archlinux TU



Vdebug



herecura.eu



Links

Vagrant multi node swarm (alpine Based just for fun)

<https://github.com/BlacklkeEagle/alpine-swarm>

Magento 2 demo with some docker image Building

<https://github.com/BlacklkeEagle/swarm-sample-magento2>