

D A T A
S Y S

Workshop Docker 101

Izolace procesů – historie

https://en.wikipedia.org/wiki/Operating-system-level_virtualization

chroot

1982

Izolace běhu DNS serveru Bind od OS kvůli bezpečnosti

FreeBSD jali nebo sysjail

2000

Izolace procesů od OS ve *BSD systémech

Bezpečnost Apache, FTP – ochrana před hacknutím v prostředí internetu

OpenVZ

2005

Open Virtuozzo je projekt řešící izolaci prostředí pro Linux
Obdoba Solaris Containers a LXC

Docker

2013

Docker je software project poskytující technologii koncept zvaný "containers" vyvinuty společností Docker, Inc.



Linux Kernel

Kontejnery

Jsou koncept userspace - uživatelského prostoru, který využívá vlastností několika subsystému linuxového jádra.

CGroups

Namespaces

Security



Klíčové prvky linuxových kontejnerů

Process Isolation

Izolace běžících procesů v kontejnerech
Izolace knihoven nutných pro běh aplikace

01

Resource Management

Poskytování zdrojů CPU, RAM, HDD od
hostovaného OS

02

Security

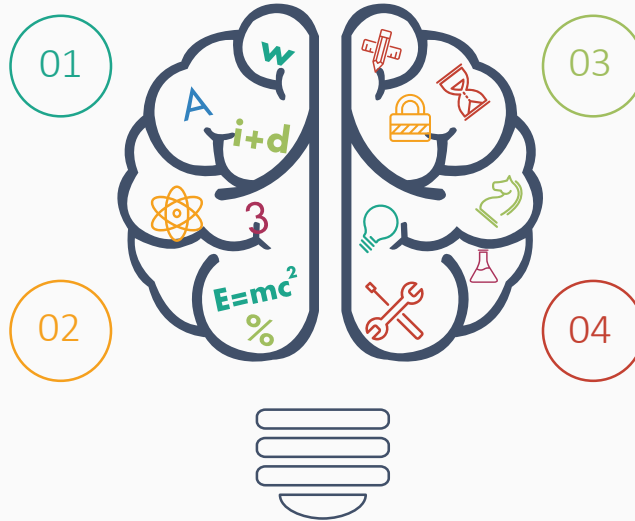
Bezpečnost v podání SELinux tad.

03

Management

Řízení běhu kontejneru start/stop atd.

04



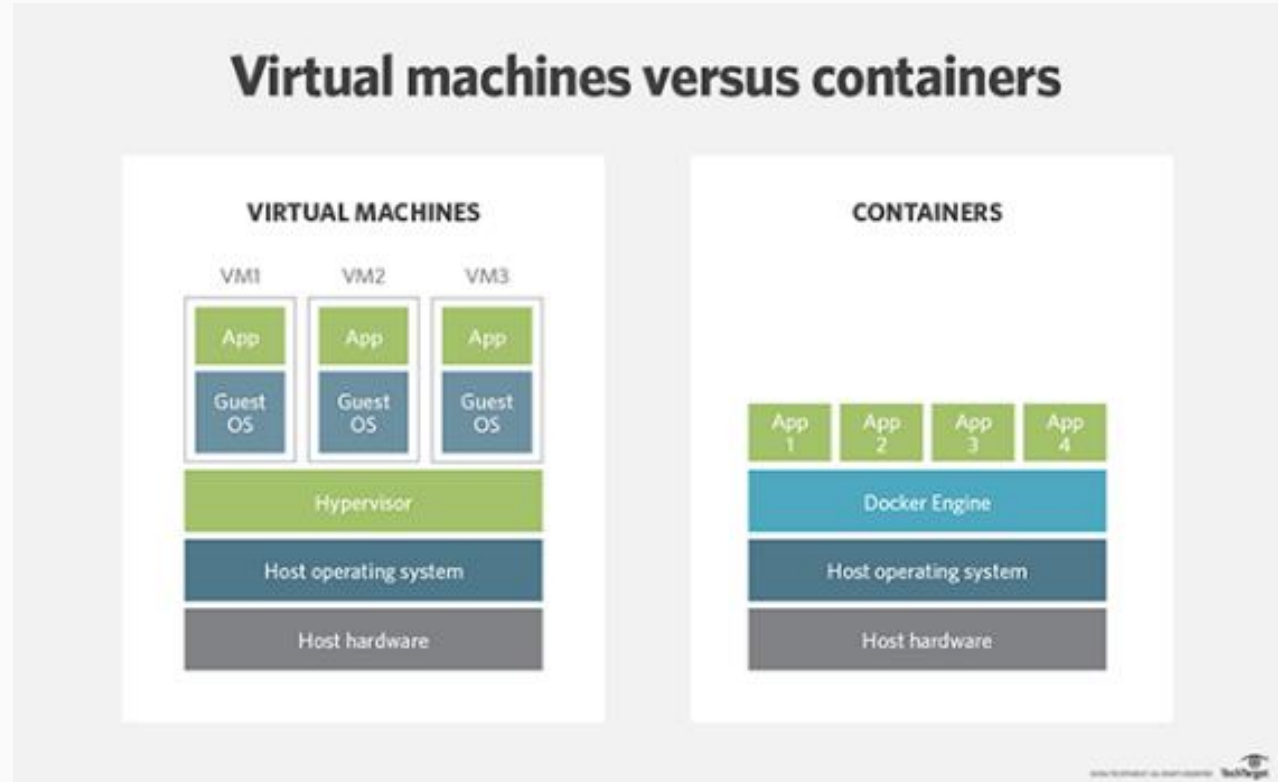
Kontejner není virtualizace

Kontejner = izolace aplikací

Aplikace běží uvnitř kontejnerů

Kontejnery provozujeme na:

Bare metal nebo Virt/Cloud



Docker kontejner - DevOps

Dev

Developer concerns

Code

Libraries

Services

Configuration

Data

All servers look the same

Ops

Ops concerns

Moving containers

Starting/Stopping containers

Logging

Monitoring

Network configuration

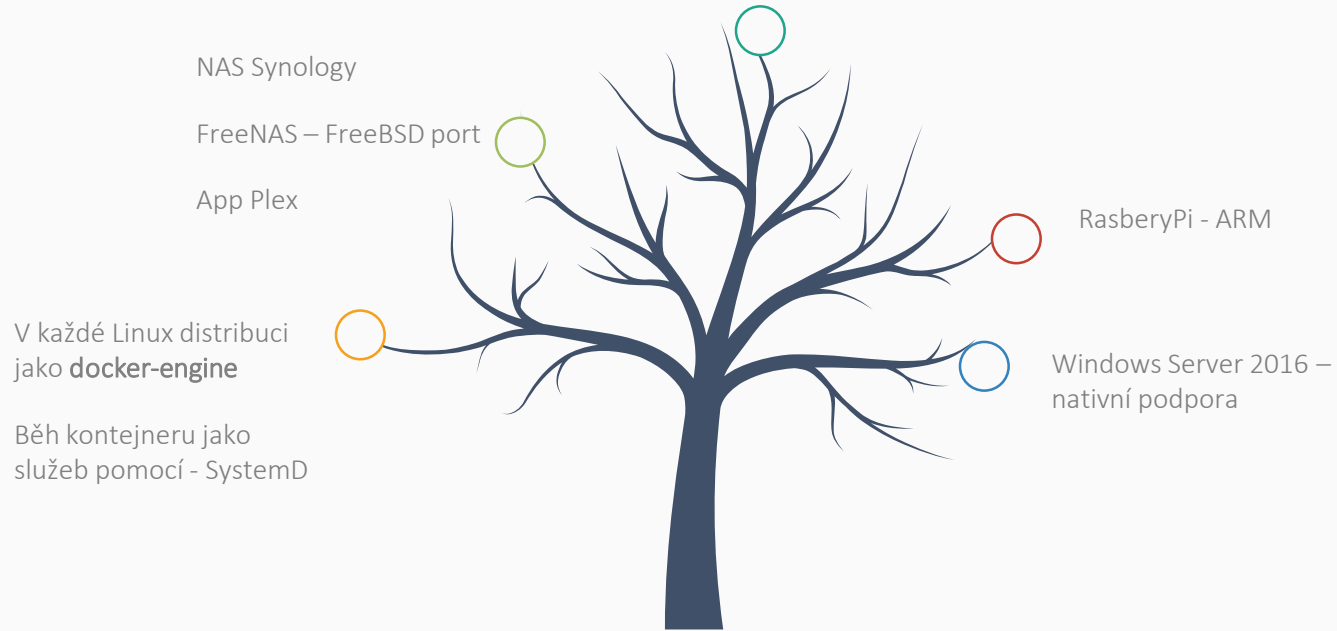
All containers look the same

Izolace

Kde dnes Docker najdeme?

Cloud - <https://cloud.docker.com>

AWS, Rackspace, Digital Ocean, Microsoft Azure, IBM SoftLayer, Packet, Vscale.io atd.



Kde najdeme Docker image?

01

<https://hub.docker.com>

Veřejná Docker Registry

02

<https://store.docker.com>

Docker obchod

03

Dockerfile



Docker slovníček

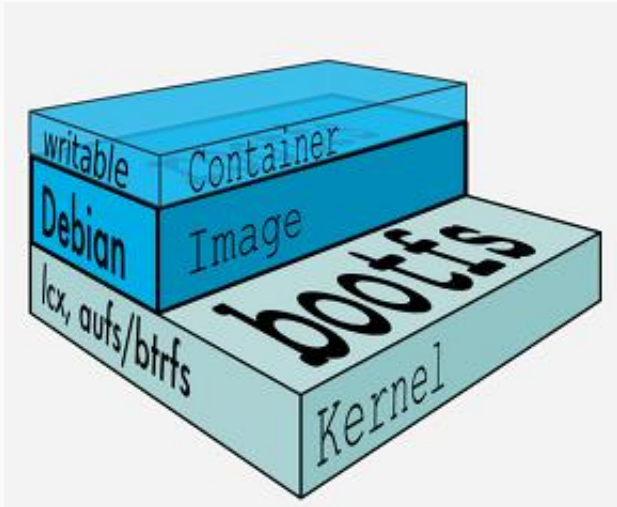
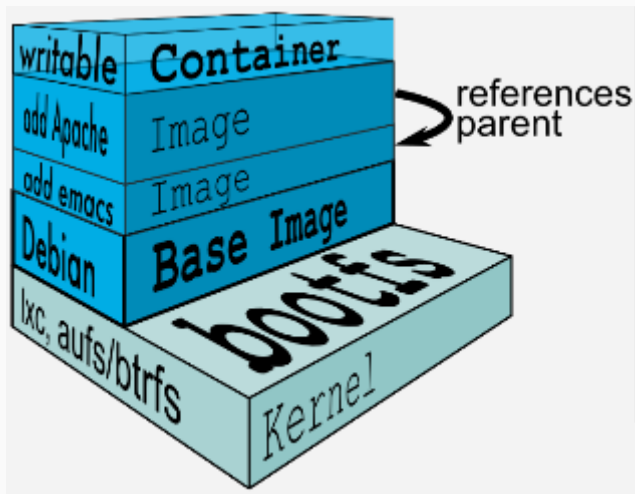


Image / Obraz

- Read-only šablona pro vznik kontejneru
- Obsahuje všechny soubory potřebné pro běh aplikace
- Má další metadata
- Binary to start

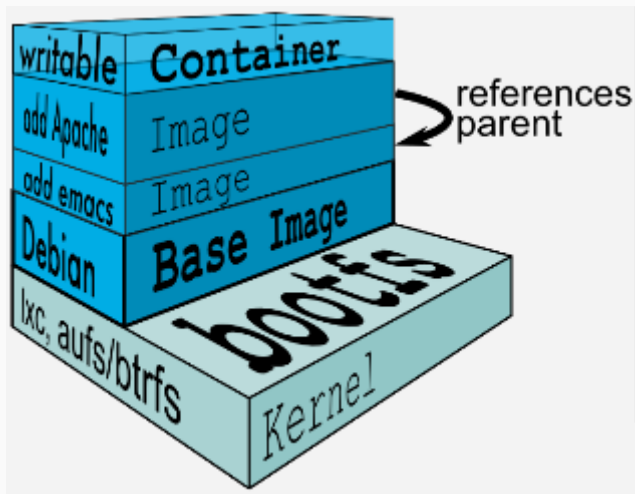
Docker slovníček



Container / Konteiner

- Běžící processes
- Na základě konkrétní Image
- Obvykle jeden process
- Izolace od hostovaného systému
- Může zapisovat do filesystému
- Commit vytvoří novou Image

Docker slovníček



Layers / Vrstvy

- Images jsou založené na rodiči – base image
- Vrstvy jsou uloženy nahoře
- Soubory v základních vrstvách jsou sdíleny mezi images
- Každý commit kontejneru vytvoří vrstvu
- Base image nemá rodič

Docker 101



První HTTP kontejner – Dockerfile

```
FROM httpd:2.4-alpine
MAINTAINER "Lukas Maly" <maly@datasys.cz>

# Copy webapp files
COPY ./public-html/ /usr/local/apache2/htdocs/

# Volume configuration
VOLUME ["/usr/local/apache2/htdocs/"]
```

Docker 101



Build - Dockerfile

```
docker build -t webserver-httpd .  
Sending build context to Docker daemon 46.08 kB  
Step 1 : FROM httpd:2.4-alpine  
--> c93eb3112b6c  
Step 2 : MAINTAINER "Lukas Maly" <maly@datasys.cz>  
--> Using cache  
--> a55ec26035a5  
Step 3 : COPY ./public-html/ /usr/local/apache2/htdocs/  
--> 7d97592fee81  
Removing intermediate container 69e89851f1d9  
Step 4 : VOLUME /usr/local/apache2/htdocs/  
--> Running in 3efe4265f23e  
--> d36e4dbe54bd  
Removing intermediate container 3efe4265f23e  
Successfully built d36e4dbe54bd
```

Docker 101



Run - kontejner

```
docker run --rm --name web-app01 \  
  -p 8080:80 \  
  webserver-httpd
```

```
docker run -d --name web-app01 \  
  -v "httpd_data":/usr/local/apache2/htdocs/ \  
  -p 8080:80 \  
  webserver-httpd
```

Docker 101



Docker Volume

```
docker volume ls  
local          httpd_data
```

```
ls -l /var/lib/docker/volumes/httpd_data/_data/  
index.html  
logo_apache.png
```

Orchestrace kontejnerů



Docker Compose

Docker Swarm – Swarm mode

Kubernetes

Amazon ECS

Azure Container Service (ACS)

Cloud Foundry's Diego

Apache Mesos - Mesosphere Marathon

Google Container Engine

Cloud Native Computing Foundation

RedHat OpenShift (Kubernetes)

Tectonic – CoreOS (Kubernetes) a CoreOS Fleet

Docker 101 – Docker Compose



HA example

```
cat docker-compose.yml
```

```
hello:
```

```
  image: tutum/hello-world
```

```
  environment:
```

```
    VIRTUAL_HOST: localhost
```

```
proxy:
```

```
  image: jwilder/nginx-proxy:alpine
```

```
  volumes:
```

```
    - "/var/run/docker.sock:/tmp/docker.sock:ro"
```

```
  ports:
```

```
    - 80:80
```

```
    - 443:443
```

Docker 101 – Docker Compose



HA example

```
docker-compose up
```

```
docker-compose scale hello=6
```

```
docker-compose kill
```

```
lynx http://localhost
```

Docker 101 – Docker Compose



Wordpress

```
cat docker-compose.yml  
version: '2'
```

```
services:
```

```
  db:
```

```
    image: mysql:5.7
```

```
    volumes:
```

```
      - db_data:/var/lib/mysql
```

```
    restart: always
```

```
    environment:
```

```
      MYSQL_ROOT_PASSWORD: my_pw_sql!
```

```
      MYSQL_DATABASE: wordpress
```

```
      MYSQL_USER: wordpress
```

```
      MYSQL_PASSWORD: wp_secret_pw!
```

```
....
```

Docker 101 – Docker Compose



Wordpress

```
...  
wordpress:  
  depends_on:  
    - db  
  image: wordpress:latest  
  ports:  
    - "8000:80"  
  restart: always  
  environment:  
    WORDPRESS_DB_HOST: db:3306  
    WORDPRESS_DB_NAME: wordpress  
    WORDPRESS_DB_USER: wordpress  
    WORDPRESS_DB_PASSWORD: wp_secret_pw!
```

```
volumes:  
  db_data:
```

Docker 101 – Docker Compose



Wordpress

```
docker-compose up -d
```

```
http://localhost:8000  
admin  
MpTktKYSZoVQJsGZ3S
```

Docker 101 – Swarm mode



<https://circularo-ds.all.local>

Circularo – DS Files

docker node ls

| ID | HOSTNAME | STATUS | AVAILABILITY | MANAGER STATUS |
|-----------------------------|--------------|--------|--------------|----------------|
| 4vtfummjkhvh4sw0hq3izxxii * | circularo-01 | Ready | Active | Reachable |
| 9nhr5u4w53qoy02tdqqn9lgap | circularo-05 | Ready | Active | |
| okyu4ipjv8pogp2jwsmmk60v1 | circularo-03 | Ready | Active | Leader |
| qpjd7acasloafx5stt4akw42s | circularo-02 | Ready | Active | Reachable |
| t7reaooqw47upixdstvlegsjh | circularo-04 | Ready | Active | |

Docker 101 – Swarm mode



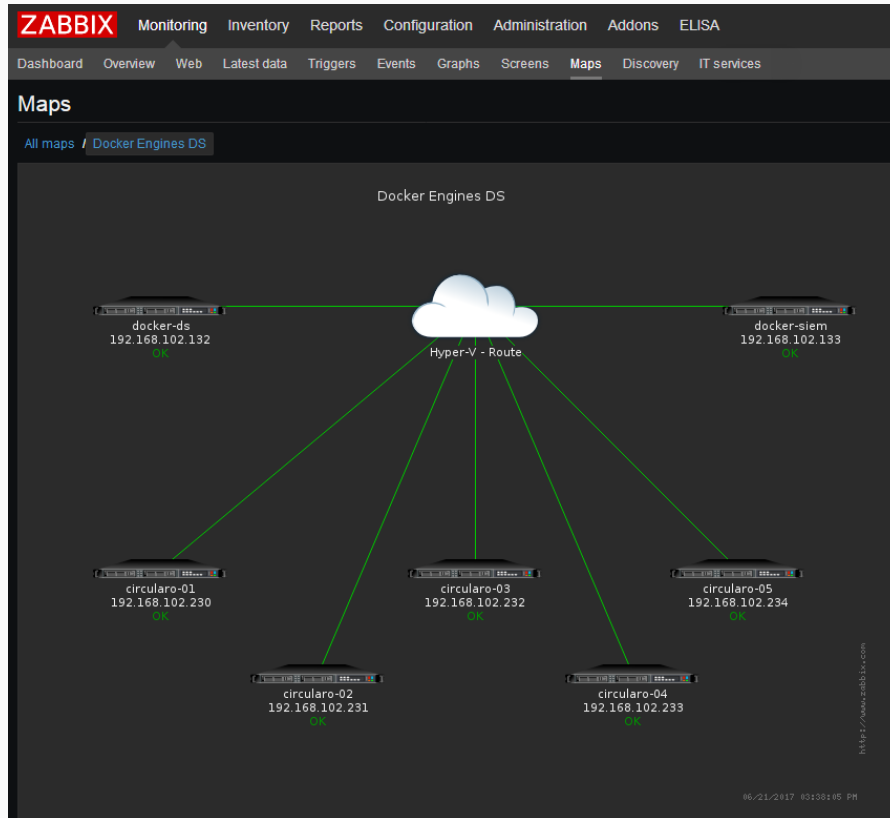
<https://circularo-ds.all.local>

Circularo – DS Files

docker service ls

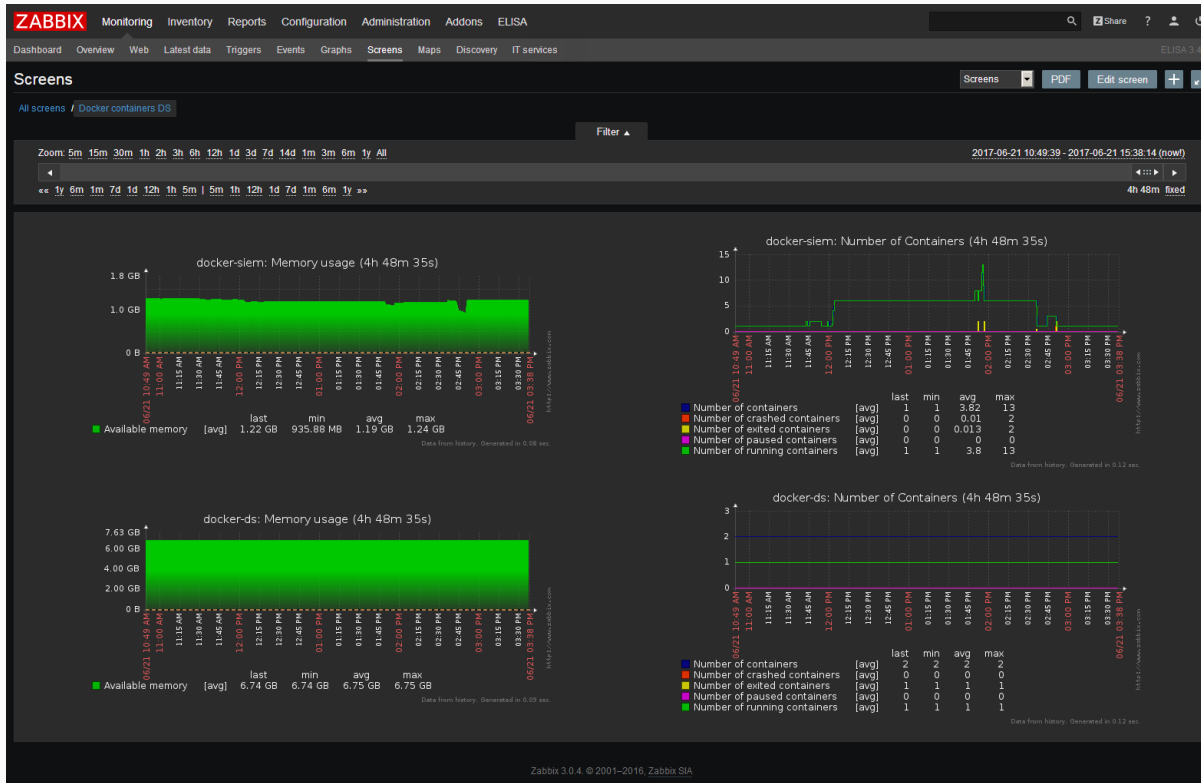
| ID | NAME | MODE | REPLICAS | IMAGE |
|--------------|-----------------------|------------|----------|---|
| 2qz0zifa1x4o | circularo-backend | replicated | 3/3 | registry-datasys.circularo.com/backend:datasys |
| 5oczoh541syx | elasticsearch-cluster | global | 4/4 | elasticsearch:2.4 |
| 9kz1yxqeukr4 | circularo-utils | replicated | 3/3 | registry-datasys.circularo.com/utils:17.1.6 |
| vpeo3melksl3 | circularo-frontend | replicated | 1/1 | registry-datasys.circularo.com/frontend:datasys |
| wj4plfeldzq9 | riak-cluster | global | 5/5 | basho/riak-kv:latest |

Zabbix Monitoring



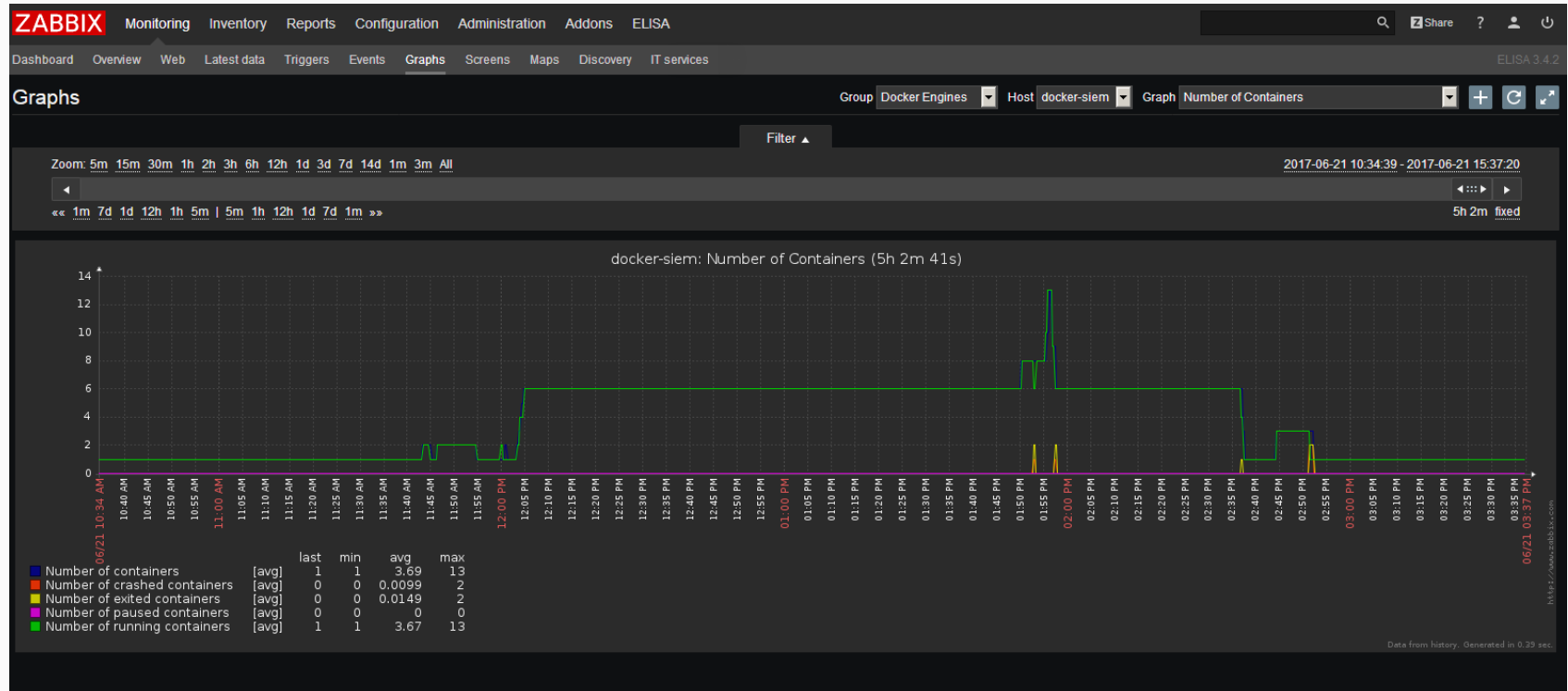
Zabbix Monitoring

<http://blog.smejdil.cz/2017/03/jak-monitorovat-docker-zabbixem.html>



Zabbix Monitoring

<http://blog.smejdil.cz/2017/03/jak-monitorovat-docker-zabbixem.html>



Děkuji za pozornost

D A T A
S Y S