stocard

ERNW
providing security.

Docker & Security

Florian Barth, barth@stocard.de
Matthias Luft, mluft@ernw.de

stocard

ERNW
providing security.

#whoweare

- Founding members of the CTF team `squareroots`
- Long-time infosec/IT nerds
- Florian: CTO @stocard.de
- Matthias: CEO @ERNW Research

stocard

ERNW
providing security.

Agenda

- Basics & Tech Stack
- Security Aspects
- (Potential) Benefits?
- Architectural Implications
- Hands-on
- Challenges
- Dev/Deployment Lifecycle meets Security

What is Docker?

- Linux-based container solution
- Used to use LXC
- LXC = Legacy API for cgroups and namespces
  - New development linuxcontainer
- cgroups: resource prioritization/limitation of
  - CPU
  - Memory
  - Network interfaces
- Namespaces: Isolation of system view as for
  - User IDs
  - Processes
  - Network interfaces
- Layered filesystem
- Shared Kernel!

stocard

ERNW
providing security.

Demo

stocard

ERNW
providing security.

Docker & Security?

- Security Objectives:
  - Isolation
  - Governance
    - I.e. no abuse of Docker to subvert security mechanisms, such as
      - patch management
      - software from trusted sources
      - segregation of test/prod

Virtualisierung im Vergleich

inovex

	Physikalischer Host	Virtuelle Maschine	Container
Gemeinsame Ressourcen	Teilen sich das Netz	Teilen sich die Host-Hardware	Teilen sich den Kernel
Angriffsszenario	Angriff per Netz auf offene Ports etc.	Angriff auf Hypervisor	Angriff per Syscall auf Kernel-Isolation (Namespaces, Cgroups, ...)
Schutzmaß-nahmen	Portfilter, Firewalls, Segmentierung der Netze	Guter Hypervisor	Absicherung im Containermanager, SE-Linux, Capabilitys
Aufwand der Maßnahmen	Einfach, Best Practices	Komplex, aber zentral zu managen	Vielschichtig durch relativ große Angriffsfläche

Quelle:
https://www.inovex.de/fileadmin/files/Vortraege/2015/docker-security-nils-magnus-guug-26.03.2015.pdf

stocard

ERNW providing security.

LXC

- Current version (Mar 2016): 1.10.0
- Before 1.0 (Feb 2014):
  - Containers could not be run as non-root users
  - Implicit possibility for container-root-user to break out of the container via sysfs
  - No user namespaces

stocard

ERNW providing security.

User namespaces

- Before user namespaces:
  - UID 0 in container was uid 0 in host
  - UID 1000 in container was uid 1000 in host
  - => 1:1 mapping
- With user namespaces:
  - UIDs in container can be mapped to UID range on host
  - Root in container != root on host!

stocard

ERNW
providing security.

Demo

Breakout Vectors

- Kernel Vulnerabilities
  - Shared kernel between container and host => Kernel vulnerability violates isolation
  - Attack surface: syscall interface
    - More than 600 syscalls…

stocard

ERNW
providing security.

Docker Attack Surface

- Governance: SW from trusted sources
  - Docker images from *docker pull*
- Docker as an vulnerable application:
  - Privilege escalation on the host
  - Remote code execution on the host

stocard

ERNW
providing security.

Known Attacks & Vulnerability History

Breakout

- Before 1.0: Breakout by design via sysfs
  - http://blog.bofh.it/debian/id_413
- "Containers do not contain"
  - http://opensource.com/business/14/7/docker-security-selinux
  - Devices are not namespaced:
    - */dev/mem*
    - */dev/sd** file system devices
    - Kernel Modules
  - If you can communicate or attack one of these as a privileged process, you can own the system.

Breakout

- "Shocker", Using capabilities:
  - CAP_DAC_READ_SEARCH and CAP_DAC_OVERRIDE
  - Allow to open files not only by pathname (which would be restricted to container layered filesystem), but also inodes
  - Iterate through inode numbers (/ is starting at 2) to access any file on the host.

stocard

ERNW providing security.

On the host

- Privilege escalation:
  - http://reventlov.com/advisories/using-the-docker-command-to-root-the-host
  - Docker socket was world-accessible (rw)
  - Create root-container with host-fs mounted

stocard

ERNW
providing security.

On the host

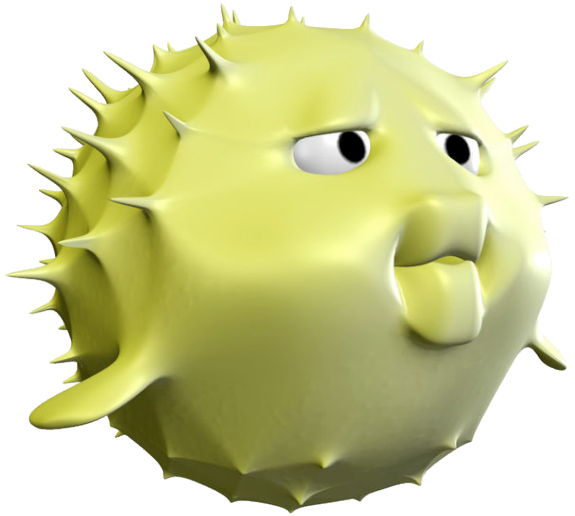
- Remote code execution
  - Symlink attacks via downloaded docker bundles that are extracted

stocard

ERNW
providing security.

Counter Measures & Hardening

The Basics...



- Docker is ”IT + X”.
- Don’t ignore traditional controls such as
  - High patch level
  - Isolation of management interfaces
  - Least privilege
- The following slides only contain Docker specifics.
- CIS Benchmark most comprehensive source (see sources).

Linux Containers

... and Hardening

- User namespaces (see above) and non-root containers
- *seccomp*: Restrict available syscalls
- *cap-drop*: Drop capabilities for the container (such as to access files based on inodes)

stocard

ERNW
providing security.

Docker improvements

- Read-only sysfs/procfs
- Command line options for *cap-drop*

Hardening Options

Docker specifics

- Use SElinux enforcement (many distribution ship proper profiles)
- Use hardened host kernel (GRsec)
- Use non-privileged containers
- Use docker-bench-security to check for security best practices
  - https://github.com/docker/docker-bench-security

stocard

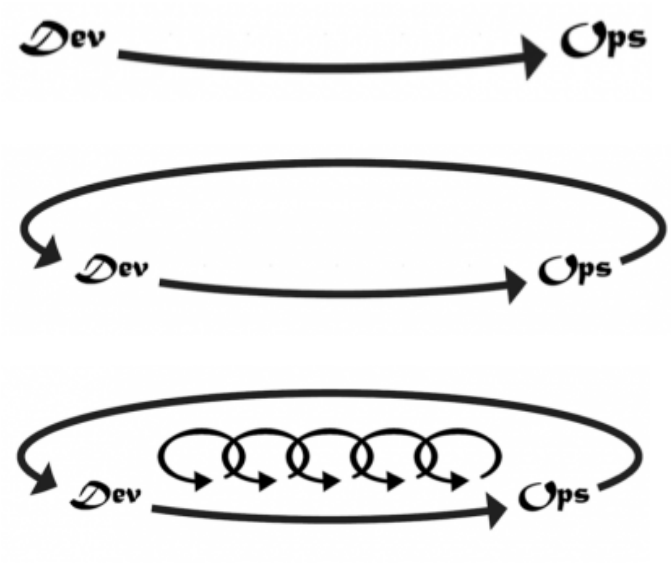
ERNW
providing security.

Benefits & Architecture


I DONT ALWAYS TEST MY CODE
BUT WHEN I DO, I DO IT IN PRODUCTION
Troll.me


WORKED FINE IN DEV
OPS PROBLEM NOW
memegenerator.net

What is DevOps?



- Culture / Mindset
- Goal: improve quality and speed at which innovation is delivered
- Embrace
  - Communication
  - Collaboration
  - Integration of Development & Operations

Micro Services

- Architectural Pattern
- “Do One Thing and Do it Well!”
- Break apart monolithic apps into micro service clusters/clouds

- e.g. Amazon, Netflix, SoundCloud

Monolith Rant

- Obstacles to frequent deploys
  - Need to redeploy *everything*
  - Long running jobs (?)
  - Increased risk of failure
- Effects
  - Infrequent updates, long QA cycles
  - Slow iterations, inhibiting experimentation
  - Slows down development
  - Communication overhead
  - Locked into tech stack

Micro Service Tribute

Benefits:

- Smaller, more understandable apps
- No dependency hell
- Reduced startup times
- Smaller & faster deploys
- Fine-grained scaling
- Fast & Reproducible tests
- No tech lock-in
- Fault isolation

… but beware

- Complexity cost
  - Deployment
  - Overhead
  - Monitoring
  - Implicit interfaces
  - Service discovery / routing
  - Shared state
- Need to build devops XP & skills

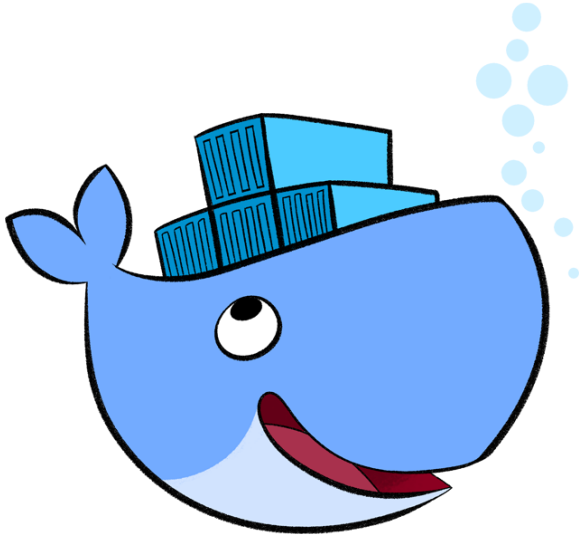
12 Factor App

- Codebase in VCS
- Dependencies explicit and isolated
- Configuration vs Code
- Backing Services
- Build, release, run
- Stateless, isolated processes
- Disposability
- Dev/Prod/* Parity
- Logs as event streams

Docker?

- Content agnostic
- Hardware agnostic
- Content isolation & interaction
- Automation
- Highly efficient
- Separation of duties

Docker Engine

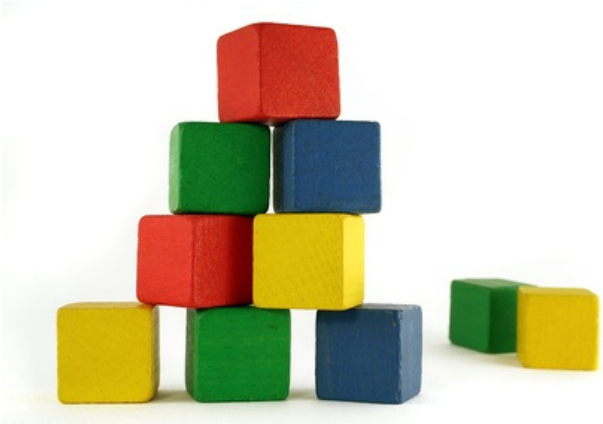


- Core of docker platform
- Offers baseline services to create and operate container
- Plugin-friendly
  - Networking
  - Logging
  - Volumes
  - Event Stream

stocard

ERNW
providing security.

Anatomy of a dockerized app



- Dockerfile – describes one service
- Images – runtime environments
- Containers – instance of app
- Volumes – non-ephemeral data
- Networks – communication

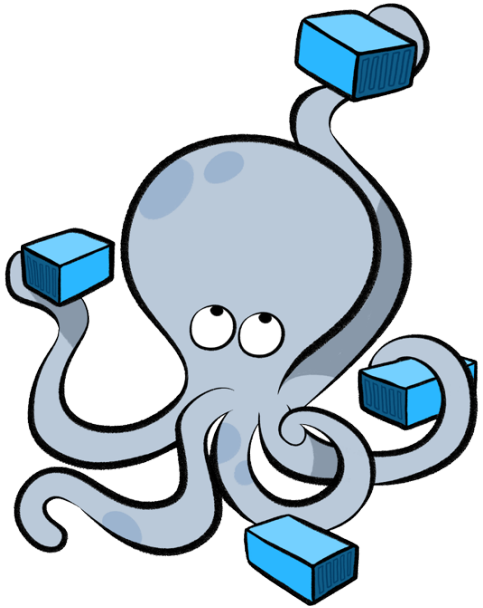
Dockerfile

DEMO

Docker Registry

- Distribution of docker images
- CI / CD stores images in registry
- Docker Engines pull images and run them

Docker Compose

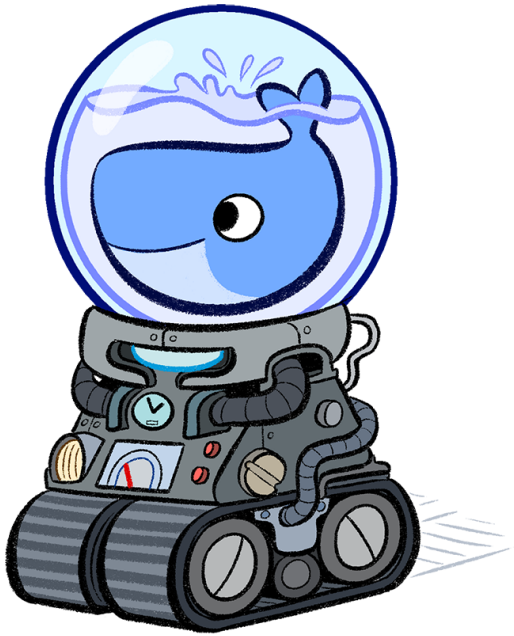


- Definition and running of multi-container applications
- yaml-based definition of your application, including:
  - Images, services, network, volumes, ...
- docker-compose up - app running!

Dockerfile

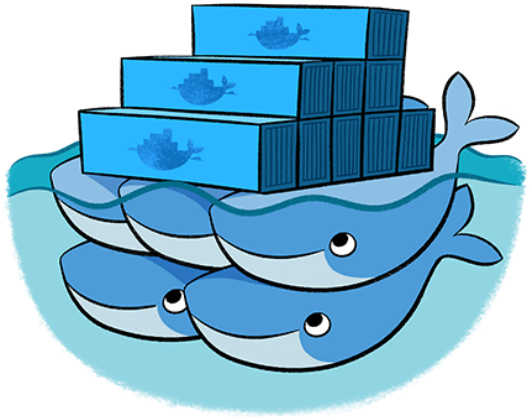
DEMO

Docker Machine



- Automatic provisioning
  - Provision host
  - Install docker engine
  - Setup secure communication
- Provisioning Drivers
  - virtualbox, bare metal, AWS, GCE, Digital Ocean, Azure, and many more

Docker Swarm



- Built-in clustering tool for docker
- Combines a pool of hosts into one virtual docker host
- Discovery Services (consul, etcd, ZooKeeper)
- Basic filtering and scheduling

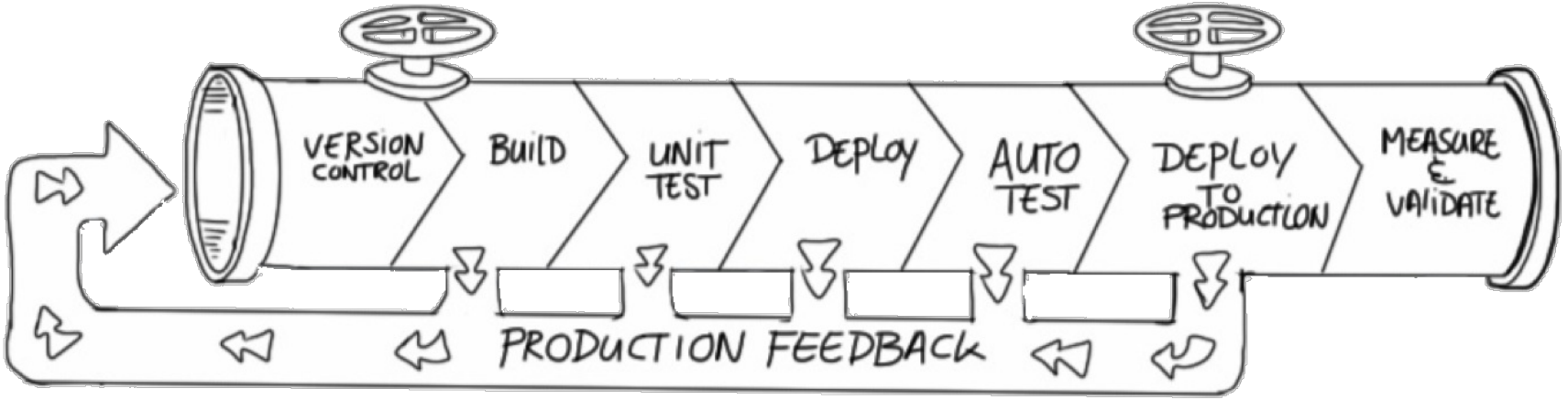
Challenges

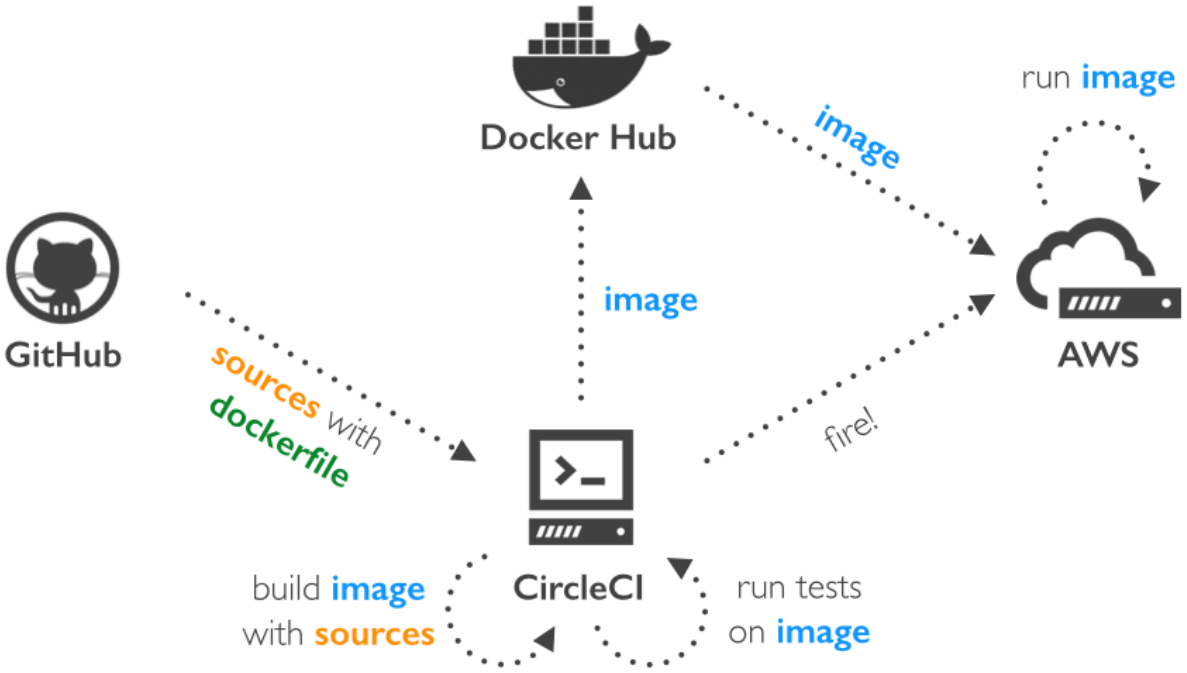
- CI/CD
- Configuration/Orchestration Management
- Service Routing
- Log Management & Monitoring

stocard

ERNW
providing security.

CI / CD – Concept

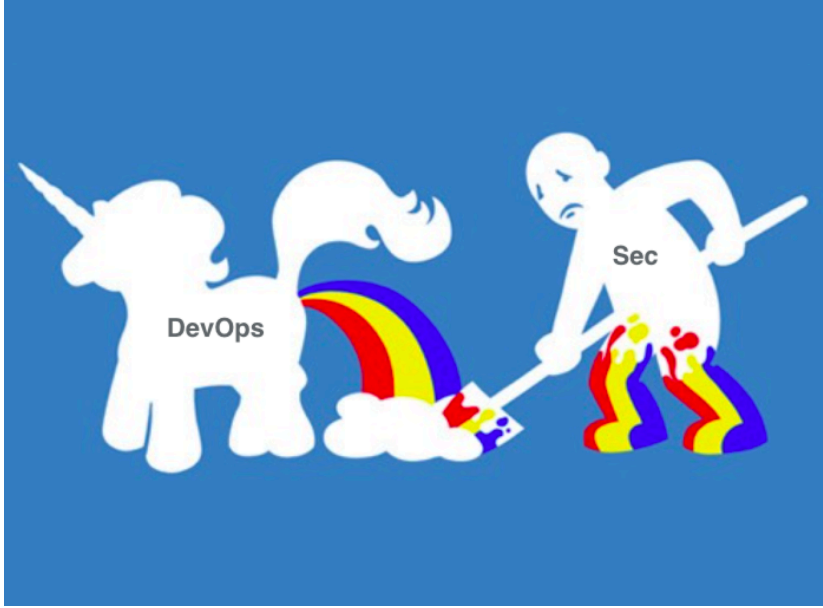




CI / CD

Instantiation

stocard



SecDevOps?

What is SecDevOps?

- Movement to make security work "DevOps" as well.
  - Haven't seen a good implementation yet.
- More interesting question:
  - How can we integrate "security" into the described CI/CD/DevOps approach?

Objectives

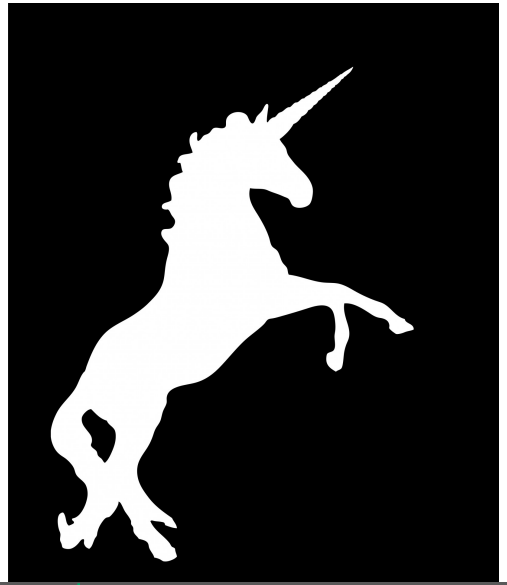
- Security of the deployed application
- Security of the docker host OS + container OS

stocard

ERNW
providing security.

Traditional Approach

If it only would be...



- Security concept at beginning of the project
- Security assessment before releases
- Final approval before go-live

Opportunities

- There is no such thing as “out-of-band-patch”.
  - Also not on the OS level!
- Integrate automatic assessment tools into the deployment process
  - Nothing new though
- As ITSec: Enable yourself to have a faster dialogue with the developers
  - Establish tools (e.g. issue tracker)
  - Vuln/risk rating metric – the simpler the better!
  - Business-reasonable risk recommendations

Conclusions


I have no idea what you're talking about...
...so here's a bunny with a pancake on its head.

Questions?

stocard

ERNW
providing security.

There's never enough time...

THANK YOU...

...for yours!

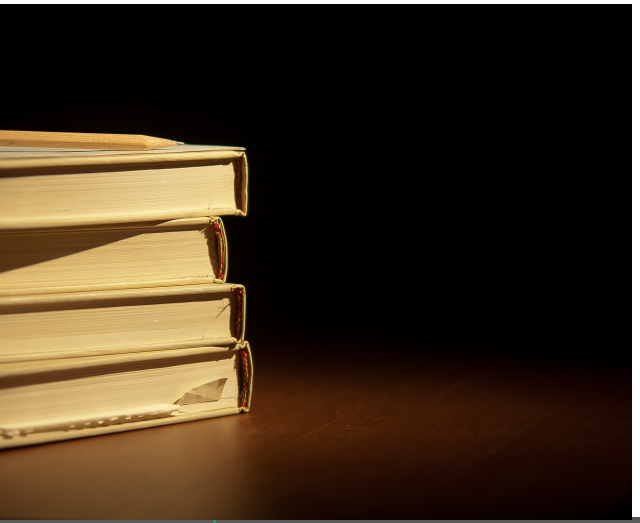
@der_cthulhu
@uchi_mata

barth@stocard.de
mluft@ernw.de

stocard

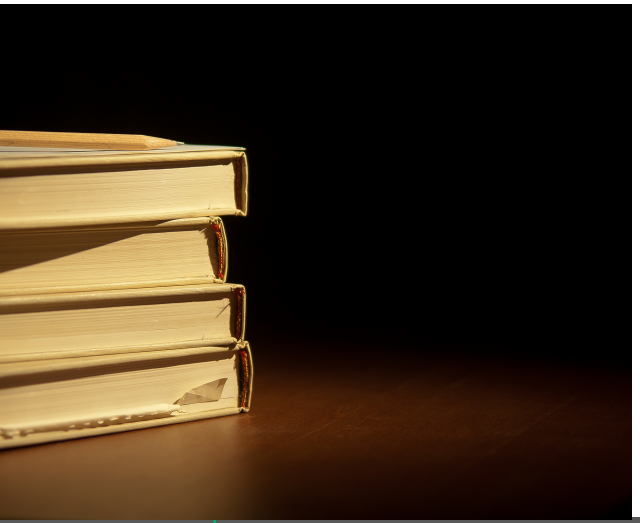
ERNW providing security.

Sources



- Docker Bench: Checking for best practices
  - https://github.com/docker/docker-bench-security
- Jérôme Petazzoni on Docker Security
  - E.g.: Containers, Docker, and Security: State of the Union
  - http://events.linuxfoundation.org/sites/events/files/slides/Containers,%20Docker,%20and%20Security_%20State%20of%20the%20Union.pdf
- http://opensource.com/business/14/9/security-for-docker
- https://zeltser.com/security-risks-and-benefits-of-docker-application/
- CIS Hardening Guide
  - https://benchmarks.cisecurity.org/tools2/docker/CIS_Docker_1.6_Benchmark_v1.0.0.pdf

Sources



- Spender, High Impact Capabilities
  - https://forums.grsecurity.net/viewtopic.php?f=7&t=2522
- http://xebia.github.io/cd-with-docker/#/
- http://www.schibsted.pl/2015/06/how-we-used-docker-to-deploy-schibsted-pl/
- http://itrevolution.com/the-three-ways-principles-underpinning-devops/