INDONESIA
OpenInfra Days

02.11.2019 | Surabaya, Indonesia

Build Secure Container Infrastructure with Kata Container

Simple Hardening for Docker Infrastructure

Yusuf Al Afid - Cloud Engineer
yusuf@btech.id

Biznet GioCloud · Biznet · Mellanox Technologies · BANK BRI · OSF OpenStack Foundation

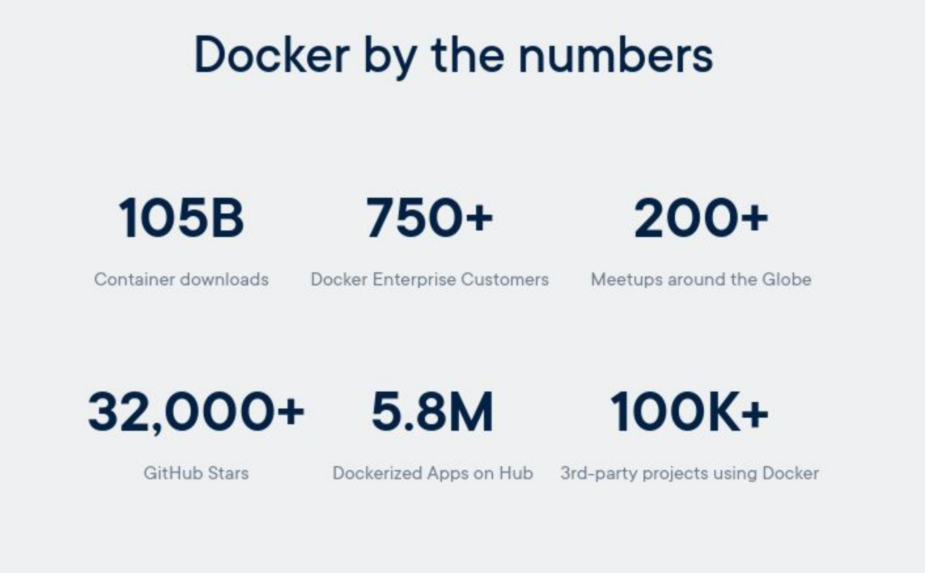
Today’s speak

Why Docker security is important?

How does Docker handle security?

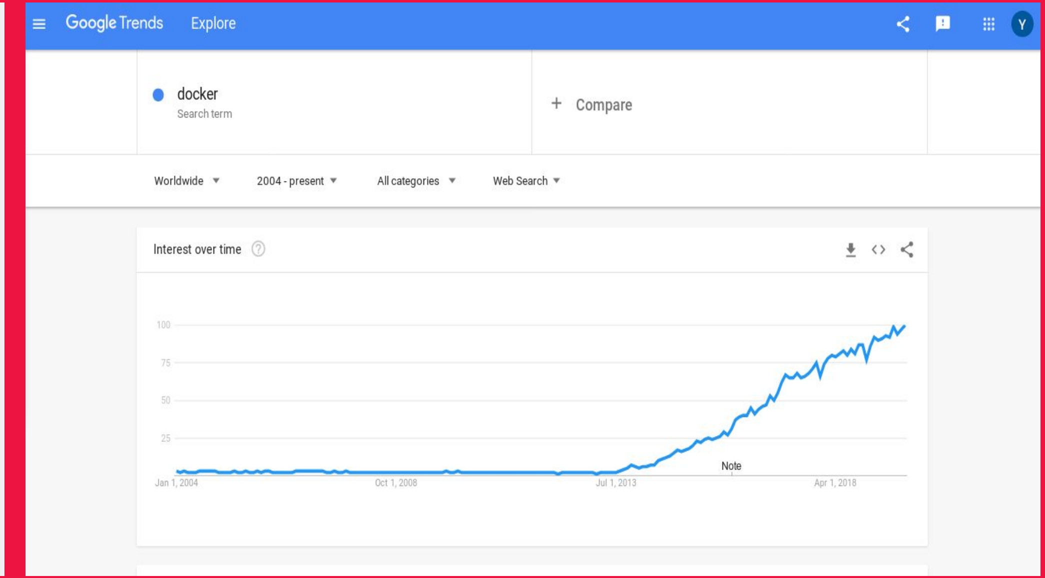
Introduction of Kata Container

Why Docker Security is important?



Docker by the numbers

105B Container downloads	750+ Docker Enterprise Customers	200+ Meetups around the Globe
32,000+ GitHub Stars	5.8M Dockerized Apps on Hub	100K+ 3rd-party projects using Docker



> “Gartner asserts that applications deployed in containers are more secure than applications deployed on the bare OS and, arguably, on a VM”

Joerg Fritsch | July 14, 2016

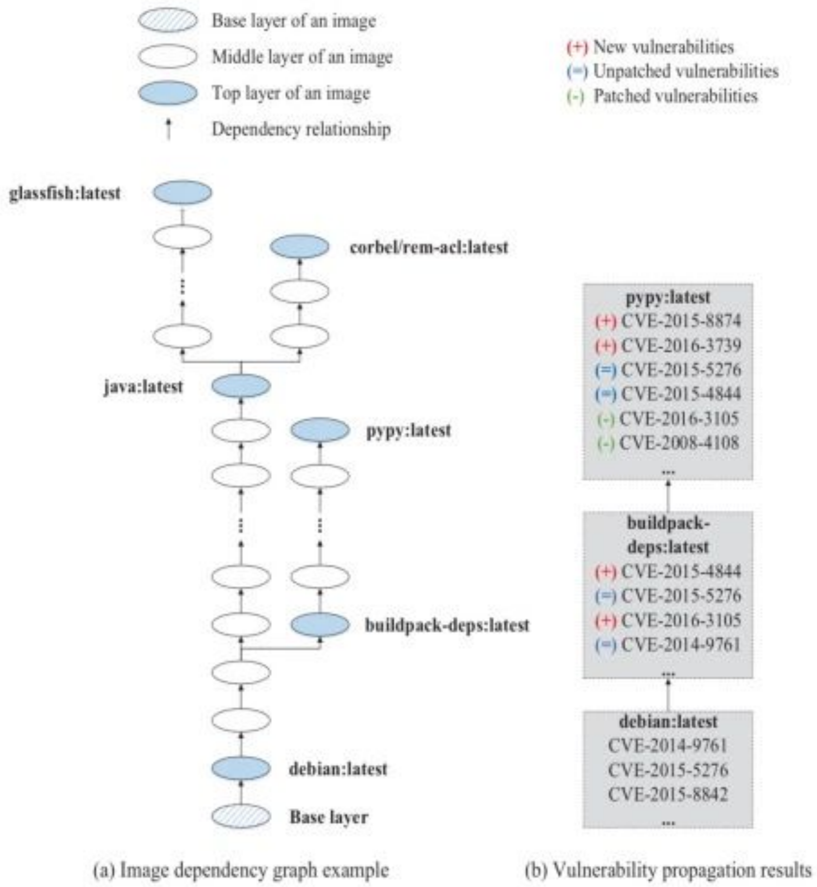
https://blogs.gartner.com/joerg-fritsch/can-you-operationalize-docker-containers/

But...

<>

Higher risk if multiple applications are run in shared host

- Escaping and Privilege Escalation
- Insecure Image
- Denial of Service
- Kernel Level Threats



(a) Image dependency graph example

(b) Vulnerability propagation results

Figure 3: Inter-image dependency analysis example.

A security vulnerability introduced at lower layers is propagated into all dependent layers

https://blog.acolyer.org/2017/04/03/a-study-of-security-vulnerabilities-on-docker-hub/

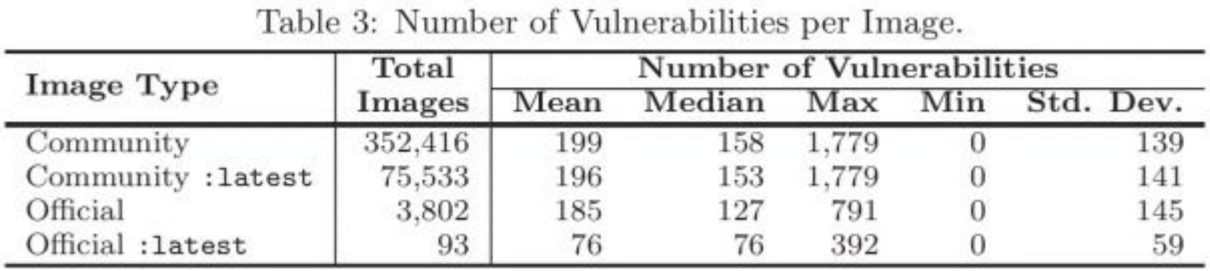


Table 3: Number of Vulnerabilities per Image.

Image Type	Total Images	Number of Vulnerabilities				
		Mean	Median	Max	Min	Std. Dev.
Community	352,416	199	158	1,779	0	139
Community :latest	75,533	196	153	1,779	0	141
Official	3,802	185	127	791	0	145
Official :latest	93	76	76	392	0	59

Docker Hub images contain ~180 vulnerabilities on average. Many images have not been updated for hundreds of days

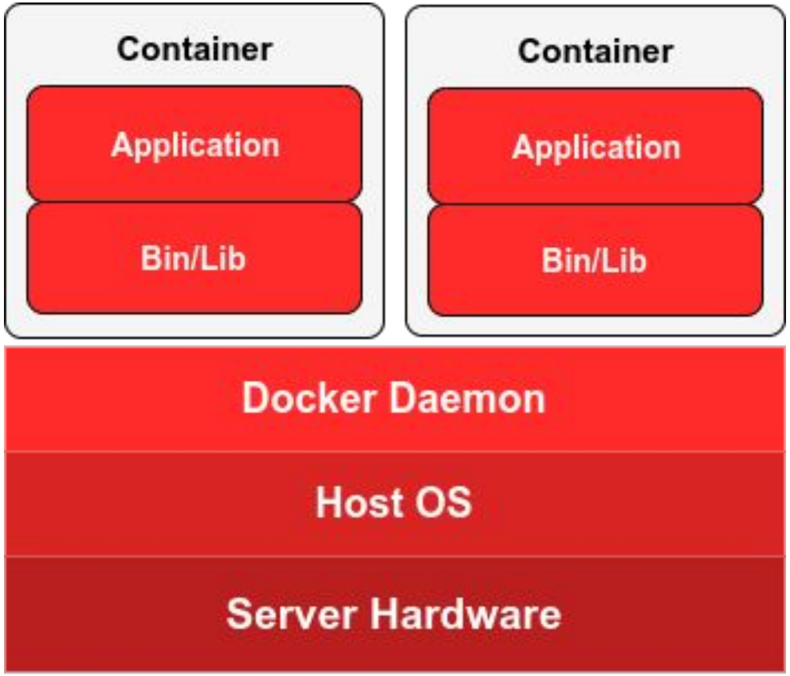
https://blog.acolyer.org/2017/04/03/a-study-of-security-vulnerabilities-on-docker-hub/

How Docker Handle Security?

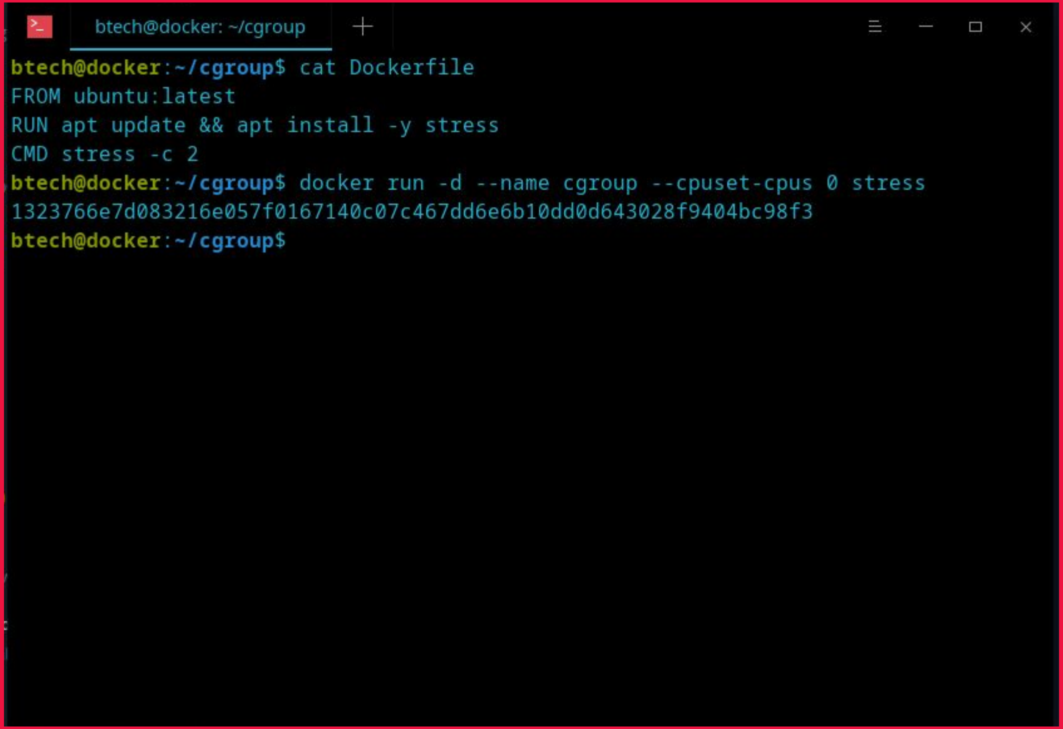
How Docker Handle Security

- ✓ Kernel Namespaces
- ✓ Control Groups
- ✓ Trusted Image
- ✓ Kernel Capabilities

Kernel Namespaces



Control Group



```
btech@docker:~/cgroup$ cat Dockerfile
FROM ubuntu:latest
RUN apt update && apt install -y stress
CMD stress -c 2
btech@docker:~/cgroup$ docker run -d --name cgroup --cpuset-cpus 0 stress
1323766e7d083216e057f0167140c07c467dd6e6b10dd0d643028f9404bc98f3
btech@docker:~/cgroup$
```

Trusted Image

✓

Pull Image with tags

Most common ways

✓

Pull image with digest

Make sure image version didn’t change

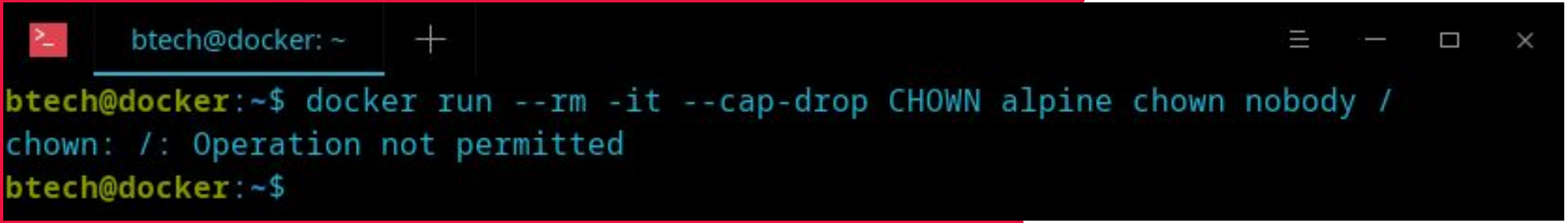
✓

Docker Content Trust

Use digital signatures for data sent to and received from remote Docker registries.

Kernel Capabilities

- Traditional UNIX systems have privileged processes (uid 0, root) and unprivileged processes (uid != 0, non-root). Root processes bypass all kernel permission checks
- In practice, if one gets into a container, limited capability possibilities make it harder to extend an attack



```
btech@docker:~$ docker run --rm -it --cap-drop CHOWN alpine chown nobody /
chown: /: Operation not permitted
btech@docker:~$
```

$ man capabilities

Increasing Docker Security

Docker Image Building

- Do not run software as root. Create an user instead
- Always build on fresh base image
- Use minimal base image
- Do not trust community images on docker hub
- Use specific version of base image
- Do not store secret into Dockerfile
- Do not install unnecessary software

Docker Runtime

- Use docker-compose instead of run container manually (multiple benefits: container linking, private network, etc)
- Drop unnecessary capabilities.

- Set read only flag
- Set memory and cpu limit

Docker Host

- Keep host kernel updated
- Use Centralized logging to monitor container logs (fluentd, splunk, etc)
- Keep Docker Update

The user who control docker daemon (docker group) effectively have root access on host

USE KATA CONTAINER

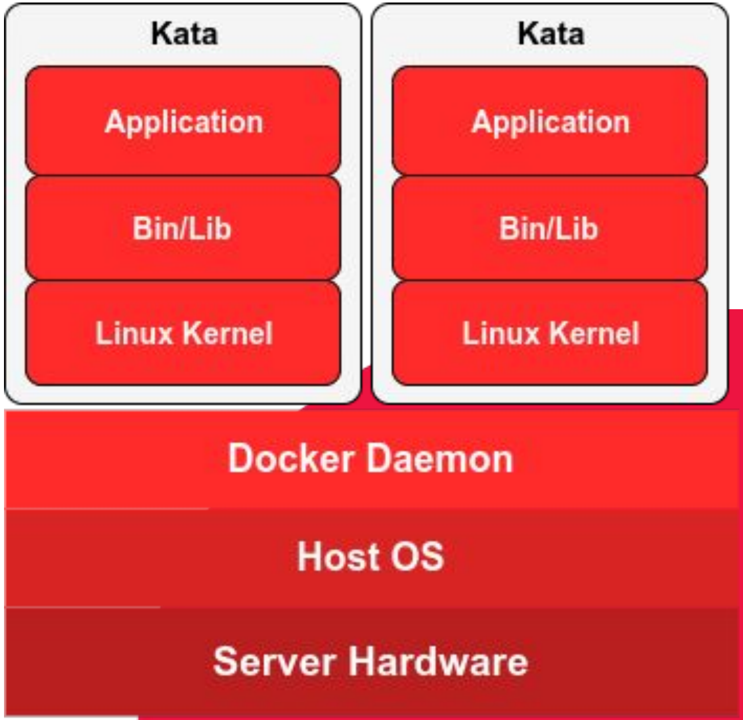
Introduction of Kata

$ history

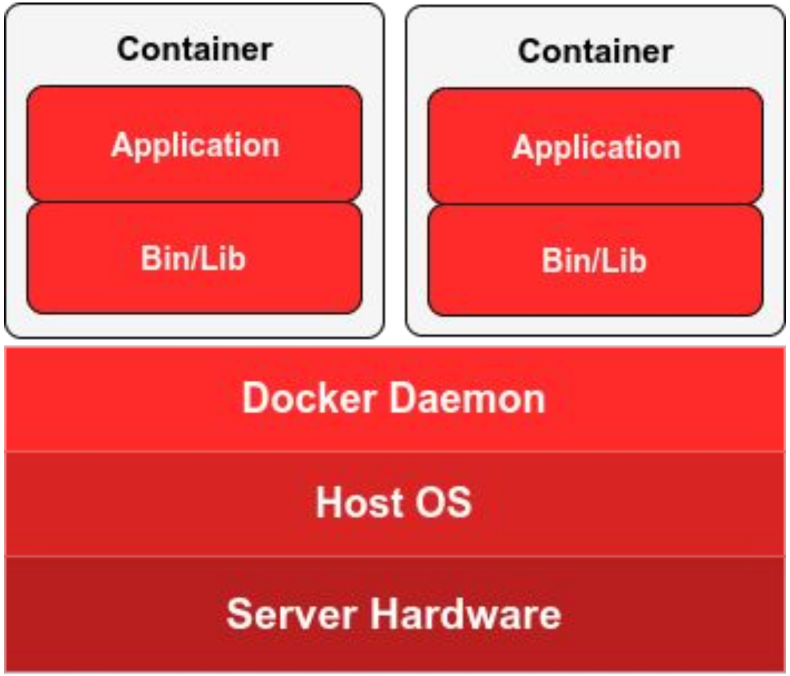
- Launch in December 2017
- Builds lightweight virtual machines that seamlessly plug into the containers ecosystem.
- Kata Containers combines technology from Intel® Clear Containers and Hyper runV to provide the speed of containers with the security of virtual machines.

Kata Container

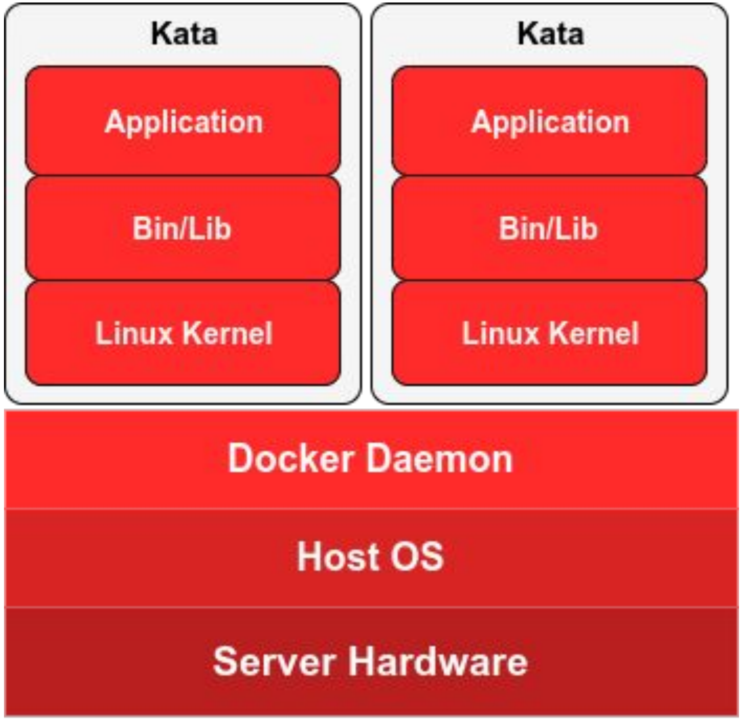
Each container/pods using hardware virtualization, to provide the speed of containers with the security of virtual machines (VMs).



Docker & Kata Container

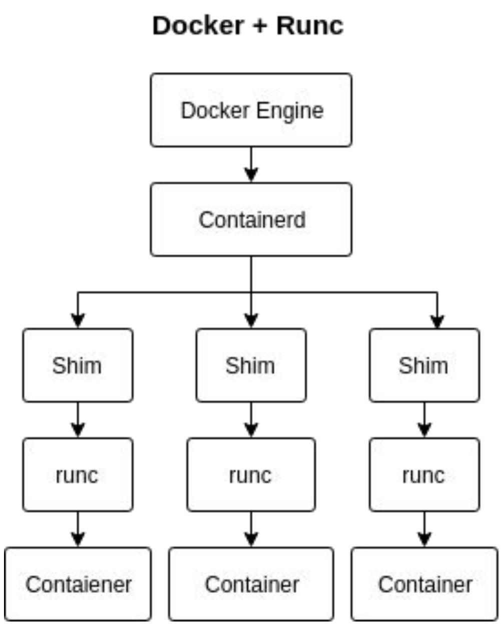


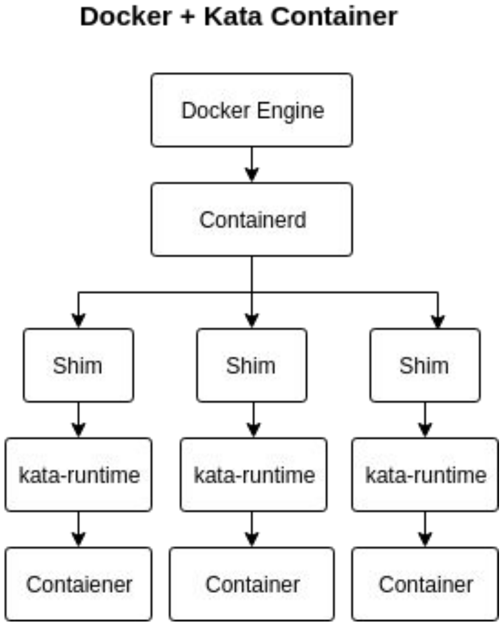
Docker



Docker with Kata

Docker & Kata Container





Thank you!