Container Orchestration is Here

What does it mean for security?

Rory McCune | July 2021

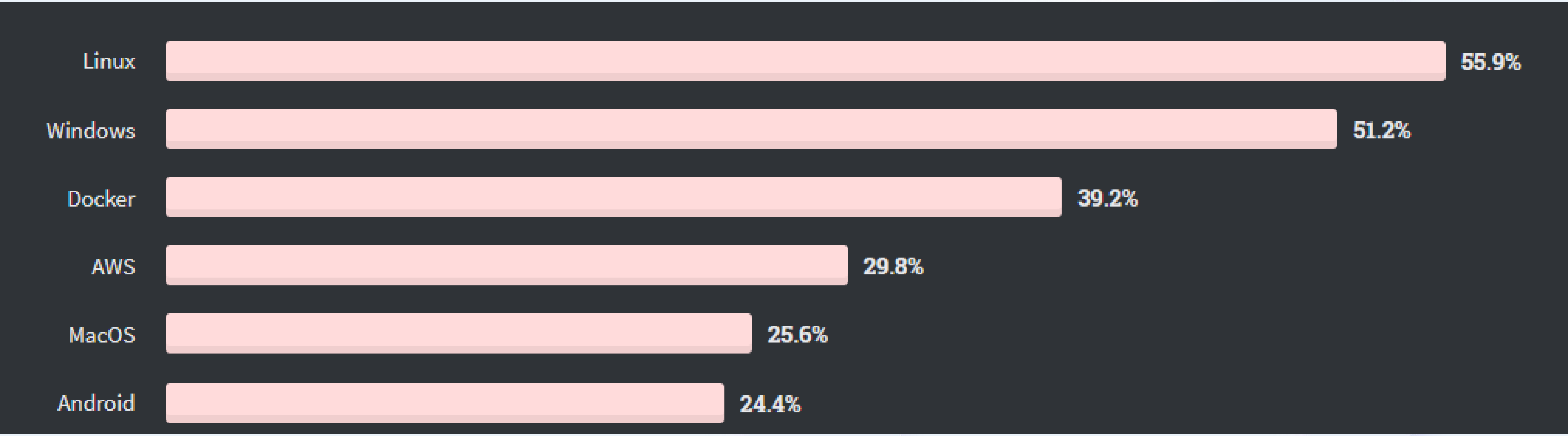
About Me

- Ex-Pentester/IT Security person
- Ex-OWASP Scotland Chapter leader
- Cloud Native Security Advocate for Aqua
- CIS Benchmark author, Docker and Kubernetes

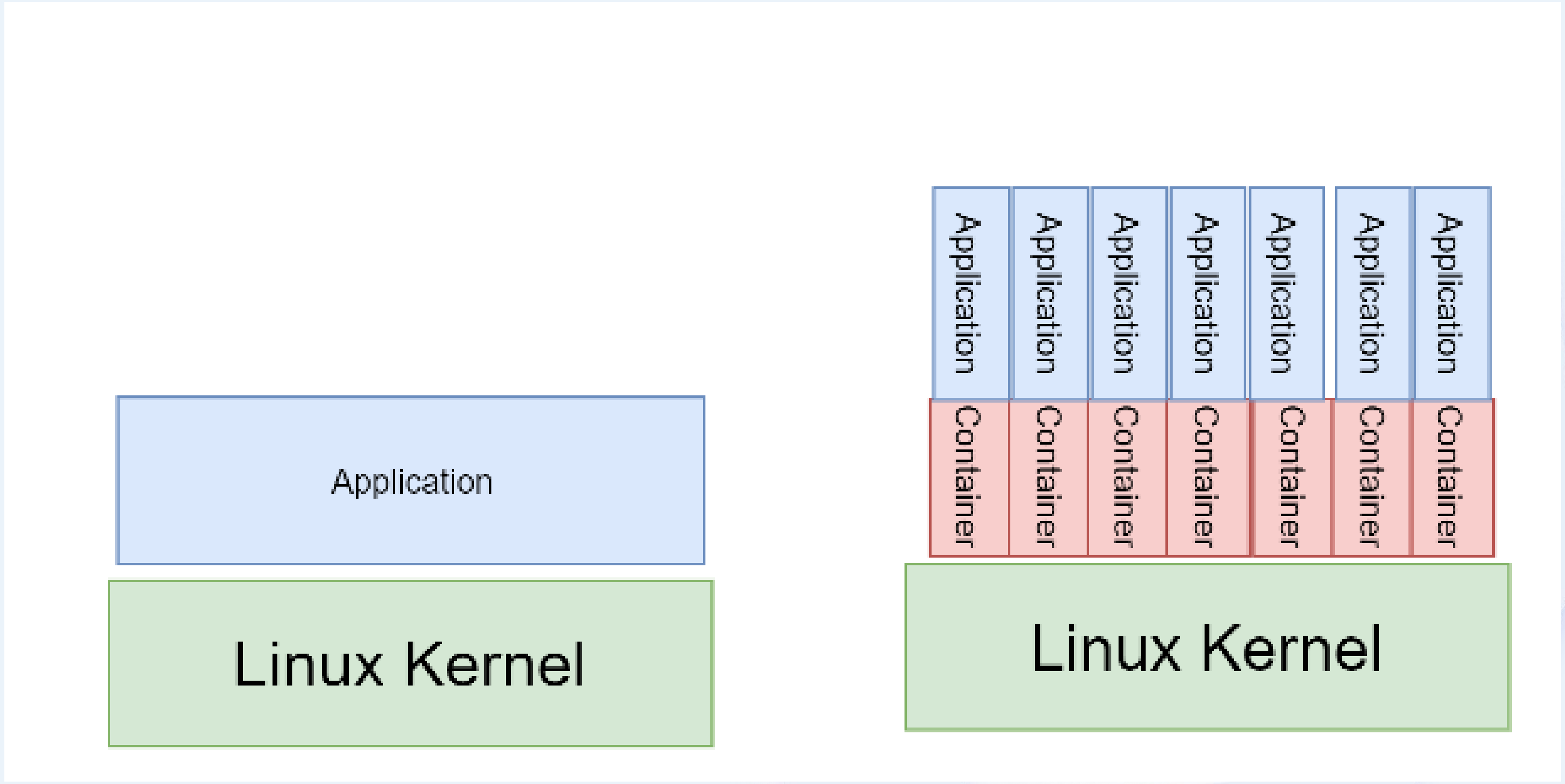
Linux Containers are not new

- 1979 - chroot system call
- 2000 - FreeBSD Jails
- 2001 - Linux Vserver
- 2004 - Solaris Zones
- 2008 - LXC
- 2013 - Docker

But they are getting quite popular..



Why are containers becoming popular? - Money



Why are containers becoming popular? – Simplicity

`docker run -d nginx` - (or any of the other 8+ million images on Docker Hub)

One Problem – Terminology Overload!

container
EKS
rolebindings
CNI
clusterrolebindings
certificatesigningrequests
customresourcedefinitions
clusterroles
roles
services
AKS
mutatingwebhookconfigurations
Kubernetes
daemonsets
deployments
GKE
istio
ingresses
nodes
networkpolicies
mesh
CRI
pods
service
validatingwebhookconfigurations
docker
serviceaccounts
podsecuritypolicies
image

What is a Linux Container?

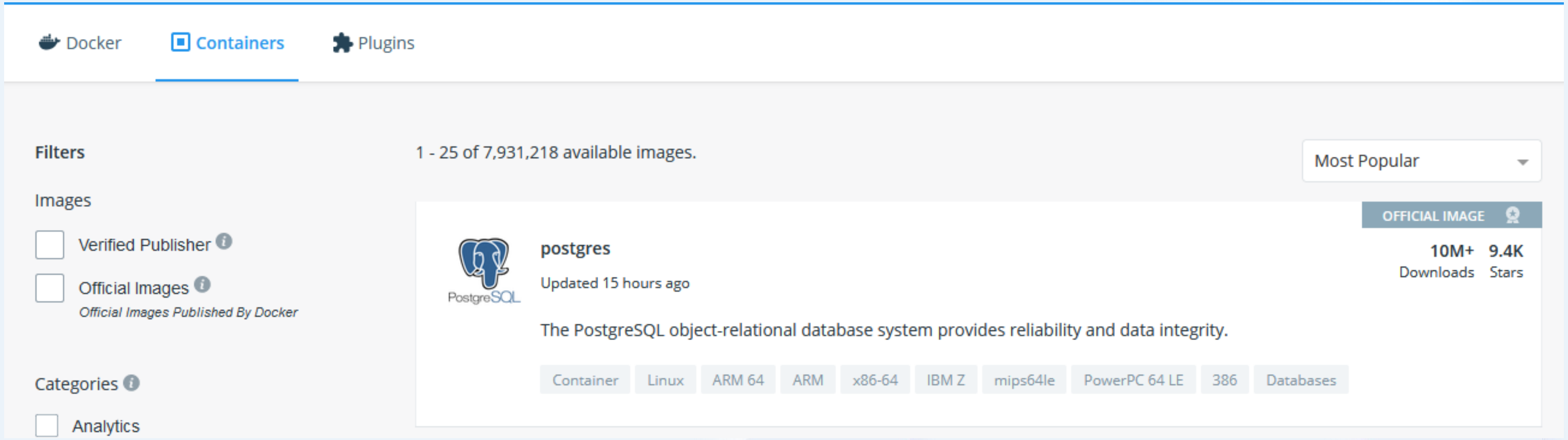
Not a virtual machine (well, usually) ☺

Linux process with an isolated view of the underlying host

- namespaces
- capabilities
- cgroups
- Apparmor/SELinux
- seccomp

Demo

So, What is a container image?


Docker Containers Plugins

Filters

Images

Verified Publisher

Official Images

Official Images Published By Docker

Categories

Analytics

1 - 25 of 7,931,218 available images.

Most Popular

OFFICIAL IMAGE

postgres

Updated 15 hours ago

10M+ Downloads **9.4K** Stars

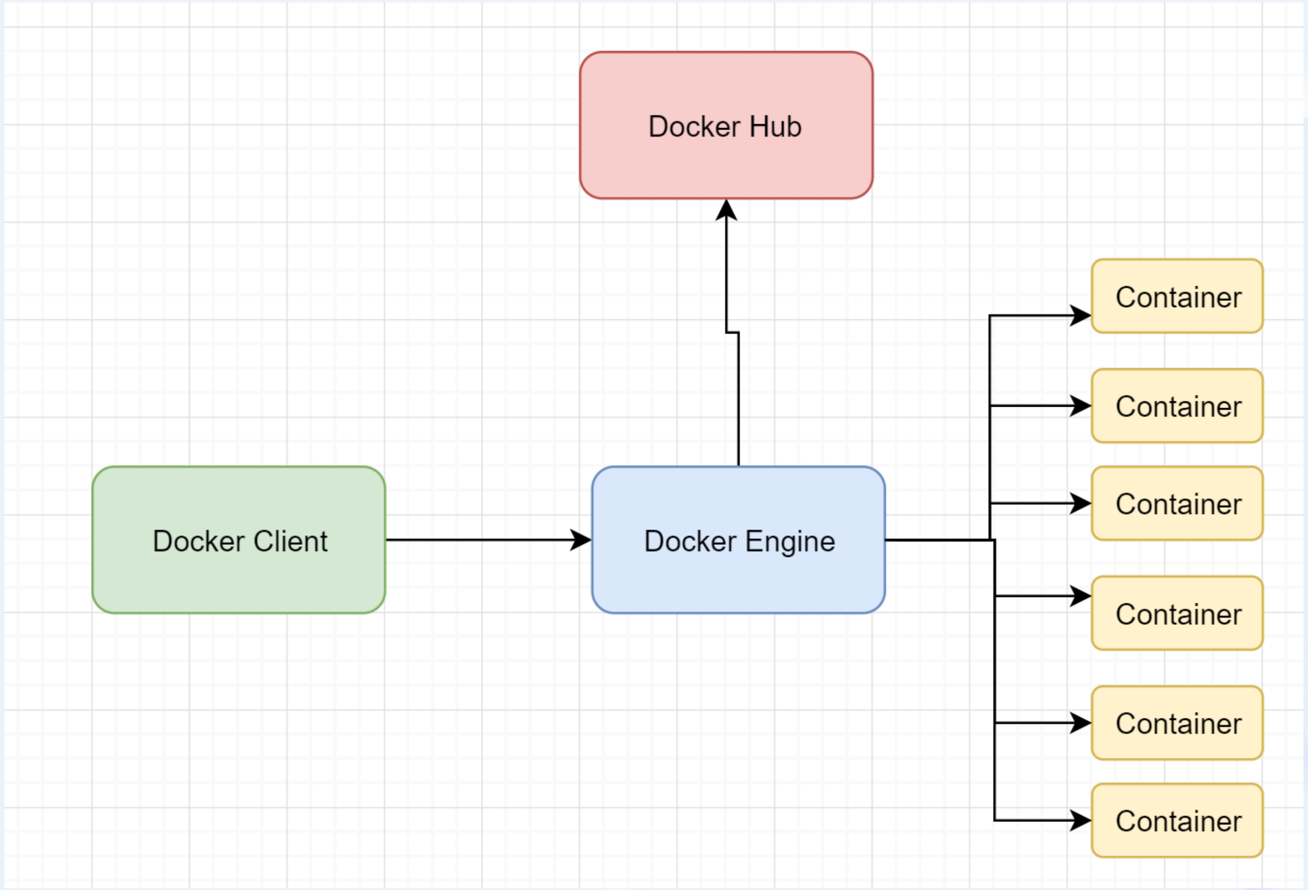
The PostgreSQL object-relational database system provides reliability and data integrity.

Container | Linux | ARM 64 | ARM | x86-64 | IBM Z | mips64le | PowerPC 64 LE | 386 | Databases

Tarball with some JSON metadata

Demo

So, What is Docker?



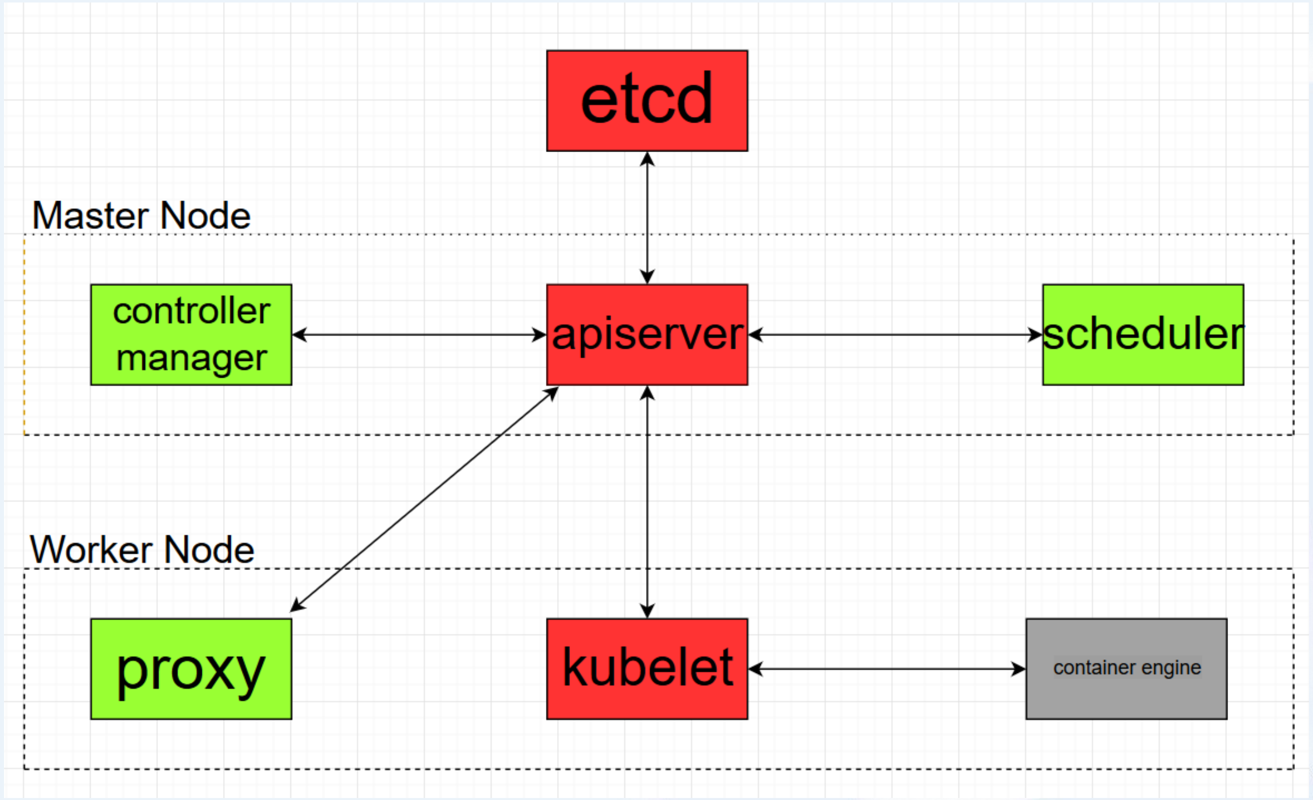
Demo

Docker == Command Execution As A Service

- Docker has a “flexible” security model
- Users with docker access should generally be considered root on the host
- “The most pointless Docker Command Ever”
-
```
docker run -ti --privileged --net=host --pid=host --ipc=host --volume /:/host busybox   chroot /host
```

Demo

So, What is Kubernetes?



Kubernetes == Distributed Remote Command Execution As A Service

- Essentially Orchestrates docker
- The same challenge applies

Demo

Securing Docker

- Docker Access == root
- Ensure images are maintained and updated
- Don't run containers as root
- Be careful giving access to host resources
- Tooling
    - Docker Bench - https://github.com/docker/docker-bench-security
    - Trivy - https://github.com/aquasecurity/trivy

aqua

Securing Kubernetes

- API Security - Authentication/Authorisation
- Security Controls
  - RBAC
  - Network Policy
  - Workload Security Policy (OPA/Kyverno/PSP)
- Tools
  - Kube-bench https://github.com/aquasecurity/kube-bench

Some Kubernetes Security Gotchas

- Authentication
    - None of the in-built Kubernetes Authentication mechanisms are suitable for Production
    - N.B. Client certificate is support but there is *no revocation*
- User Management
    - Kubernetes does not have a user database
    - There is a hard-coded cluster-admin group, system:masters
- Support Lifecycle
    - 9-12 months

Securing Containerized Environments

- Areas of differences from “traditional” environments
- Workloads are Ephemeral
  - Don’t Patch Running Containers
  - Credentials are held outside the workloads
- Everything is code
  - Infrastructure
  - Policy
  - Workloads

Conclusion

- Containerization makes heavy use of existing technologies
- A lot of existing security techniques can be re-used
- Some new approaches
- Some definite gotcha’s to look out for

Questions?

Twitter - @raesene

E-mail – rory.mccune@aquasec.com