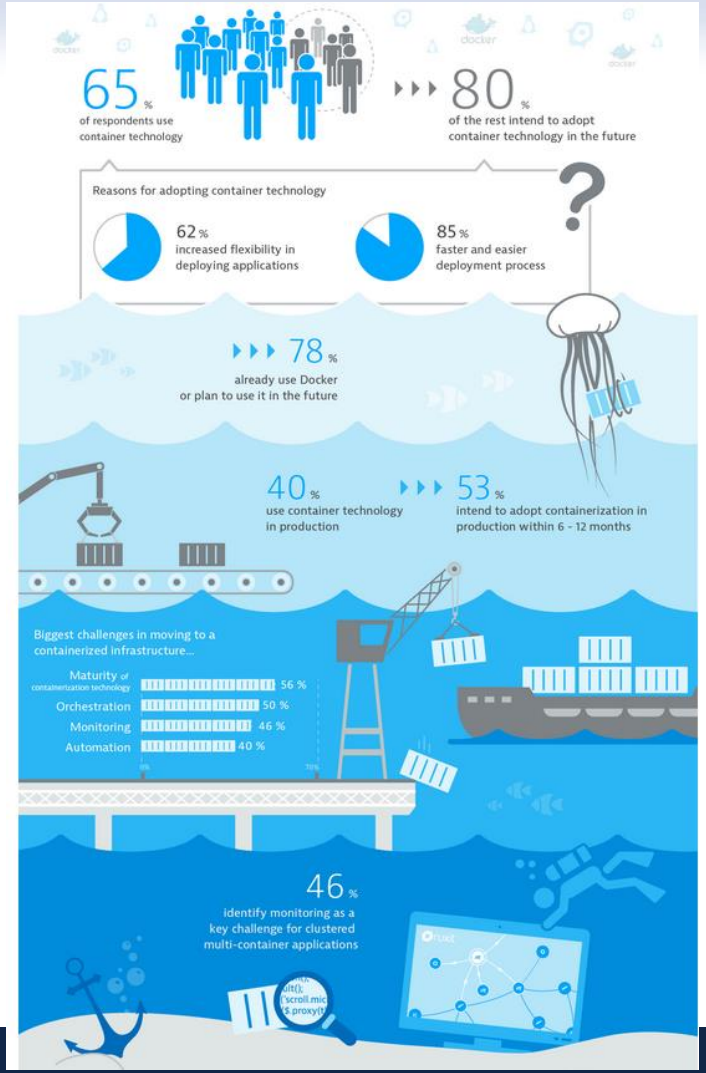

# JBug OWL
## Docker for Managers

Armin Vogt
avogt@s-und-n.de

# Wer nutzt schon Docker

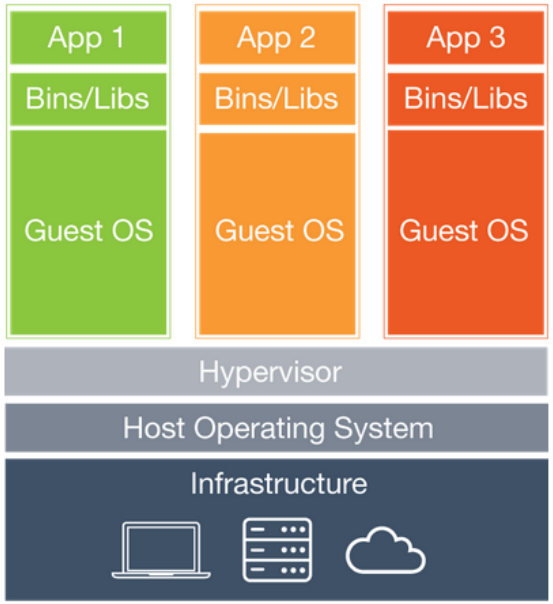
- https://www.docker.com/customers
- Docker and Containerization survey results



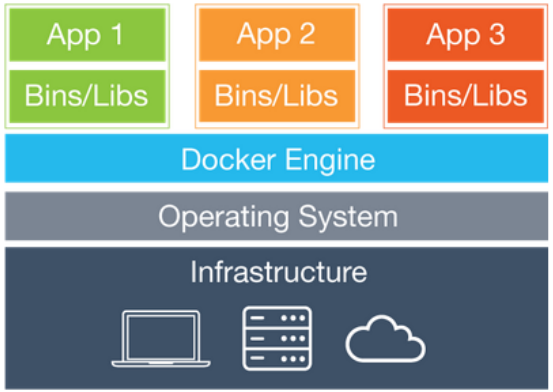
# Was ist Docker?

- rpm Pakete, die nicht installiert werden
- Laufzeitumgebung für Prozesse
- Private Netzwerkinterfaces mit Port-mapping und NAT

# Virtualisierung? Nein




Virtual Machines




Containers

# Docker = Repeatability

# „Docker ist noch zu neu!! Bleeding edge“

- Docker basiert auf Linux Containers (IBM, 2008)
  - Cgroups (2006)
    - Vserver (2001)
- Beziehung zu SELinux
  - Ziel: abgeschottete Prozesse
  - Docker ist ein Security Projekt!

# Was ist dann das „Problem“?

- Docker rührt an viele Themen, die bisher unbearbeitet blieben und zwingt zu Beschäftigung
  - Zertifikatsmanagement (eigene ROOT CA)
  - Credentials für technische User (keine Klartextpwds)
  - Verwaltung von Installationspaketen (Prozess, Zugriffsrechte)
  - Automatisierung (scripting) und ihre Überwachung
  - CIP

# CIP - CDP

- Fortwährend = engl.
  - Continous
  - Continuos
  - Continouos
  - Continuous
- CI = C.Integration  CD = C. Delivery

# Eines zum anderen

- Du willst Docker
  - Deine Firma ist hinter einem Proxy (pwd?)
    - Squid
  - Du sollst nicht soviel aus dem Internet laden!
    - Squid, Nexus als Spiegel
  - DU willst automatisiert bauen
    - Jenkins plus Docker build step plugin!
    - Der will die images ablegen…
      - Du brauchst eine Registry
        - Die braucht SSL Zertifikate, Zugänge
          - Vault, LDAP, Ldap Admin !

# Docker muss vermarktet werden!

- Weil strategische Weichenstellung Unterstützung durch das strategische Management benötigt
  - Erhebliche Arbeitsleistung erforderlich für Kick-off
  - Eingriffe in Infrastruktur müssen mit den Verantwortlichen argumentiert werden
- Andernfalls
  - Alle in einem Boot

# Wer sitzt im Boot?

- Firewall Team,
- Rechenzentrum, (virtuelle Server ?)
- Netzwerk Team (routing von container ips, VxLan)
- Vertrieb: SLA muss fixiert werden
  - Zwischen Software-Lieferant und CIP -> Quality Gates

# Rollen ändern sich

- Entwickler -> DevOps
- Administrator/Deployer -> Dev-Ops
- Operator -> OpsDev

# Docker – die Rettung für den schlampigen Entwickler

- Peter Criens, Osgi
  - Nur mit Binärkompatibilität gerettet statt portabel/adaptabel
- Antwort von Docker:
  - Wir halten die Eintrittsschwelle niedrig
    - Entwickler werden die Chance nutzen, sich selber zu optimieren
    - Kodieren statt administrieren
    - Iterieren und Einchecken
    - Automatisch Testen

# Gemeinsames Lernen - DockerHub

- „Schau sich einer diese Container an!“
  - An Stelle von Installationsanleitungen ein docker Projekt (=Code) zum ausprobieren und nachvollziehen (reengineering)

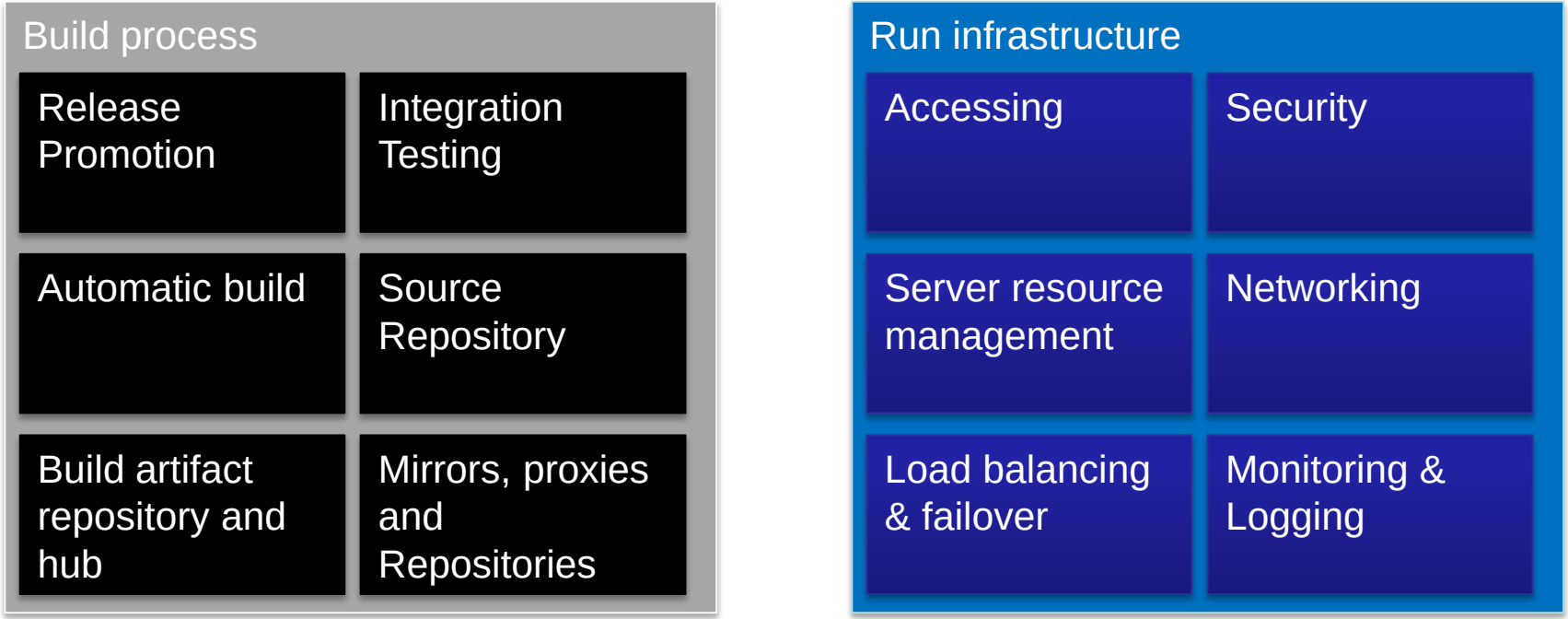
# Wie gut sind die freien Tools?

- Warum werden lieber kommerzielle Tools gekauft?
  - Weil man glaubt Support zu erhalten (und wenn das nicht reicht, kann man mit dem Finger auf ihn zeigen)
- Vergleiche Open Source:
  - Keine Telefonnummer, Support durch Entwickler, ist viel qualifizierter, aber
  - Schwerer zu bekommen
    - Sprache: a) English, b) „Entwickler“

# Endlich Kompetenz und Verantwortung in der Produktion

- Deployment kommt weg von try&error
- Bereits früher: in der Entwicklung, Teil des Contin.Build
- Die Installation wird testbar und Qualität messbar

# building blocks



# Docker – der Weg in die Cloud

- Auch wenn ein Cloud-Hosting nicht Unternehmensziel ist..
- durch Gleichförmigkeit und Entkopplung vom Host
  - Linux: alles mit Kernel >=3.x.x
  - Netzwerk: logische Netzgruppen abstrahieren von Topologie
  - Keine Root-Rechte nötig für Deployer

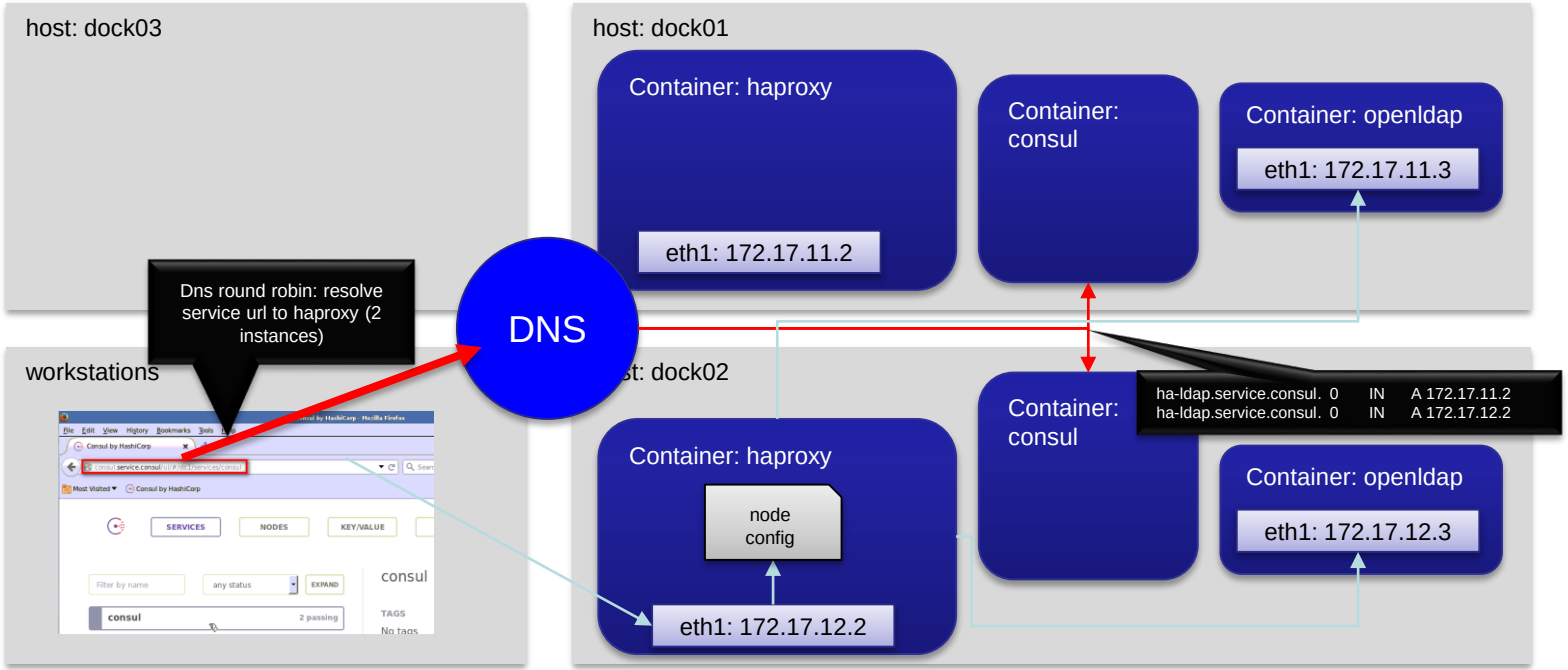
# Entwicklungsumgebungen

- Begrenzung auf eine Umgebung pro Server entfällt
  - n Abnahmeumgebungen statt einer
- Übertragbarkeit
  - In der Entwicklung beim Dienstleister können Teile der Kundenumgebung mittels Docker bereitgestellt werden
    - - inklusive Testdaten!

# Micro Services – Service Discovery - Failover

- Verschiedene Ziele, derselbe Weg!
  - Komponenten müssen auffindbar sein (late binding)

# Load balancing



# Multi-host networking

- Sicherheit, Fokussierung auf das Notwendige
  - Docker stellt immer die Abhängigkeiten in den Vordergrund
- Container arbeiten in einem benannten logischen Netzwerk zusammen
  - Dieses kann über mehrere Hosts gehen

# .. Und Red Hat?

- Fabric8
- Kubernetes
- OpenShift v3 / Origin PAAS
  - **“Standardization through Containerization”**
- Von der Plattform aus („von unten“) sehen alle Container gleich aus
- Vom Container aus sieht die Plattform gleich aus


# Vielen Dank !

Armin Vogt
**Werftarbeiter**

avogt@s-und-n.de