CONTINUOUS DELIVERY MIT DOCKER

INNOVATIVE SOFTWAREENTWICKLUNG

Gunther Bachmann
gunther.bachmann@itemis.de
xing.com/profile/gunther_bachmann

CONTINUOUS DELIVERY

"CD bezeichnet eine Sammlung von Techniken, Prozessen und Werkzeugen, die den Softwareauslieferungsprozess (englisch: Deployment) verbessern." (*WikipediA*)

- Software in kurzen Zyklen produzieren
- Stabile Releases jederzeit möglich
- Kontinuierlich bauen, testen und produktiv setzen

WARUM?

- Frühe Nutzung gebauter Software (Geschäftswert)
- Frühes Feedback , schnelle Reaktion auf …
  - … Fehlersituationen
  - … Marktbewegungen
- Risiko von Fehlentwicklungen minimieren

~~CONTINUOUS~~ DELIVERY

- Produktivsetzung 2 - 3 mal im Jahr
- Die Tests brauchen Wochen
- Die Tests sind durchsetzt mit manuellen Tätigkeiten
- Kampf von Entwicklung mit Betrieb
  - Entwicklungsumgebung ≠ Produktionsumgebung
  - Testumgebung ≠ Produktionsumgebung

DOCKER

BUILD, SHIP, AND RUN ANY APP, ANYWHERE

- Entwicklungsumgebung ≈ Produktionsumgebung
- Testumgebung ≈ Produktionsumgebung
- Automatisierter Produktivsetzungsprozess denkbar
- DevOps freundlich

DOCKER IM DETAIL

- Linux basiert (Kernel Virtualisierung)
- Service Isolation
- Komplett via CLI
- Ressourcen freundlich
- Docker Repositories
- Orchestrierungswerkzeuge

DOCKER VS. VM

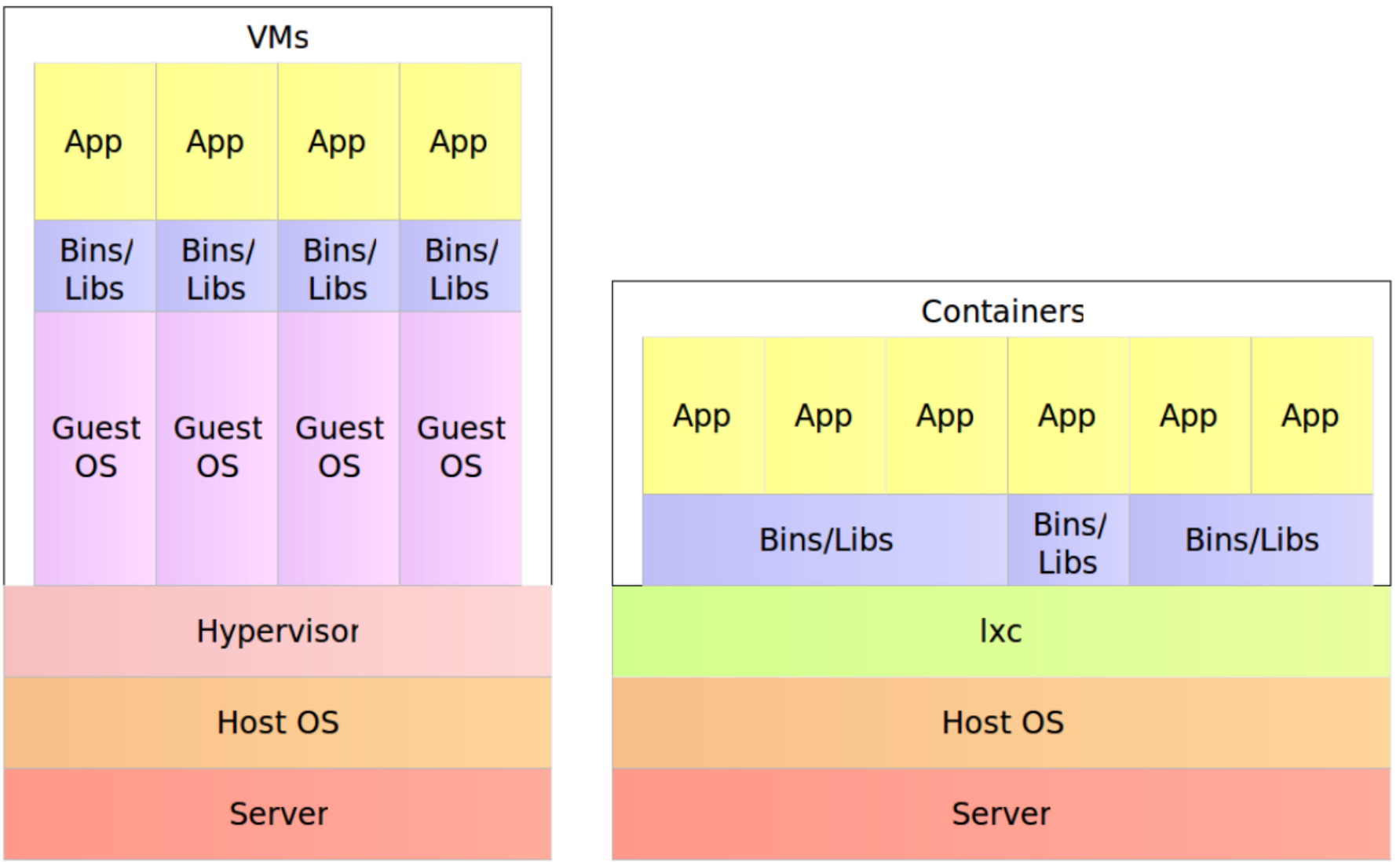


Figure 2-8: Virtual machines and containers resources utilization comparison

PERIODIC TABLE OF DEVOPS TOOLS (V2)

EMBED DOWNLOAD ADD

Os Open Source	SCM	Database Mgmt	Build
Fr Free	CI	Repo Mgmt	Testing
Fm Freemium	Deployment	Config / Provisioning	Containerization
Pd Paid	Cloud / Iaas / Pass	Release Mgmt	Collaboration
En Enterprise	BI / Monitoring	Logging	Security

1 Fm Gh Github																	2 Fm Aws AmazonWeb Services
3 Os Gt Git	4 Pd Dm DBmaestro												5 En Ch Chef	6 En Pu Puppet	7 Os An Ansible	8 En Sl Salt	9 Os Dk Docker	10 Pd Az Azure
11 Fm Bb Bitbucket	12 Os Lb Liquibase												13 Os Ot Otto	14 En Bl BladeLogic	15 Os Va Vagrant	16 Fr Tf Terraform	17 Os Rk rkt	18 En Gc Google Cloud Platform
19 Os Gl GitLab	20 En Rg Redgate	21 Os Mv Maven	22 Os Gr Gradle	23 Os At ANT	24 Os Fn FitNesse	25 Fr Se Selenium	26 Os Ga Gatling	27 Fr Dh Docker Hub	28 Os Jn Jenkins	29 Pd Ba Bamboo	30 Os Tr Travis CI	31 Pd Gd Deployment Manager	32 Os Sf SmartFrog	33 Os Cn Consul	34 Os Bc Bcfg2	35 Os Mo Mesos	36 En Rs Rackspace
37 Os Sv Subversion	38 En Dt Datical	39 Os Gt Grunt	40 Os Gp Gulp	41 Os Br Broccoli	42 Fr Cu Cucumber	43 Os Cj Cucumber.js	44 Fr Qu Qunit	45 Os Npm npm	46 Fm Cs Codeship	47 Pd Vs Visual Studio	48 Fm Cr CircleCI	49 Fr Cp Capistrano	50 Fr Ju JuJu	51 Os Rd Rundeck	52 Os Cf CFEngine	53 Fr Ds Swarm	54 Os Op OpenStack
55 Os Hg Mercurial	56 En Dp Delphix	57 Fr Sb sbt	58 Os Mk Make	59 Os Ck CMake	60 Fr Ju JUnit	61 Fr Jm JMeter	62 Fr Tn TestING	63 Os Ay Artifactory	64 Fm Tc TeamCity	65 Fm Sh Shippable	66 Os Cc CruiseControl	67 En Ry RapidDeploy	68 Fm Cy CodeDeploy	69 En Oc Octopus Deploy	70 En No CA Nolio	71 Os Kb Kubemetes	72 Fm Hr Heroku
73 En Cw ISPW	74 En Id Idera	75 Os Msb MSBuild	76 Os Rk Rake	77 Fr Pk Packer	78 Os Mc Mocha	79 En Xltv XL TestView	80 Os Jm Jasmine	81 Os Nx Nexus	82 Os Co Continuum	83 Fm Ca Continua CI	84 Pd So Solano CI	85 En Xld XL Deploy	86 En EB ElectricBox	87 Fm Dp Deploybot	88 En Ud UrbanCode Deploy	89 Os Nm Nomad	90 En Os OpenShift

91 En Xlr XL Release	92 En Ur UrbanCode Release	93 En Bm BMC Release Process	94 En Hp HP Codar	95 En Au Automic	96 En Pl Plutora Release	97 En Sr Serena Release	98 Pd Tfs Team Foundation	99 Fm Tr Trello	10 Pd Jr Jira	101 Fm Rf HipChat	102 Fm Sl Slack	103 Fm Fd Flowdock	104 Pd Pv Pivotal Tracker	105 En Sn ServiceNow
106 Os Ki Kibana	107 Fm Nr New Relic	108 Os Ni Nagios	109 Os Zb Zabbix	110 En Dd Datalog	111 Os El Elasticsearch	112 Os Ss StackState	113 En Sp Splunk	114 Fm Le Logentries	115 Fm Sl Sumo Logic	116 Os Ls Logstash	117 Os Gr Graylog	118 Os Sn Snort	119 Os Tr Tripwire	120 En Ff Fortify

XebiaLabs
Deliver Faster

Follow @xebialabs

AUTOMATISIERUNG → PIPELINE

- Das Programmieren ist (noch) nicht automatisierbar
- Das Deployment schon (Pipeline)
  1. Feature Branch baut, Unit Tests laufen, Metriken stimmen
  2. Feature Branch merged nach Develop baut, Unit/Integration Tests laufen, Metriken stimmen
  3. Deployment in das Staging, Integration/Last/Performanz/Security/Akzeptanz Tests laufen
  4. Deployment in die Produktion (inaktiv), Smoke Tests laufen, Blue/Green Deployment
  5. Rollback (optional)

AUTOMATISIERUNG VON ANFANG AN

- Von Anfang an gesamtes Deployment abbilden. Warum?
  - SW wird entwickelt, um durch die Pipeline zu kommen
  - Pipeline funktioniert und wird sukzessive erweitert
- Pipeline ist Software (Infrastructure as Code)

ZUSÄTZLICHE VORAUSSETZUNGEN

- Ausgereifter agiler Entwicklungs- **prozess**
  - Feature Branches / Pull Requests, Repository Organisation
  - Tickets, Code Reviews, Clean code, Metriken, Pipeline
- Testbarkeit
  - Software wurde **testbar** entwickelt
  - Einhaltung der Testpyramide, **Testdaten**
- Operations
  - pets vs. cattle
  - DevOps

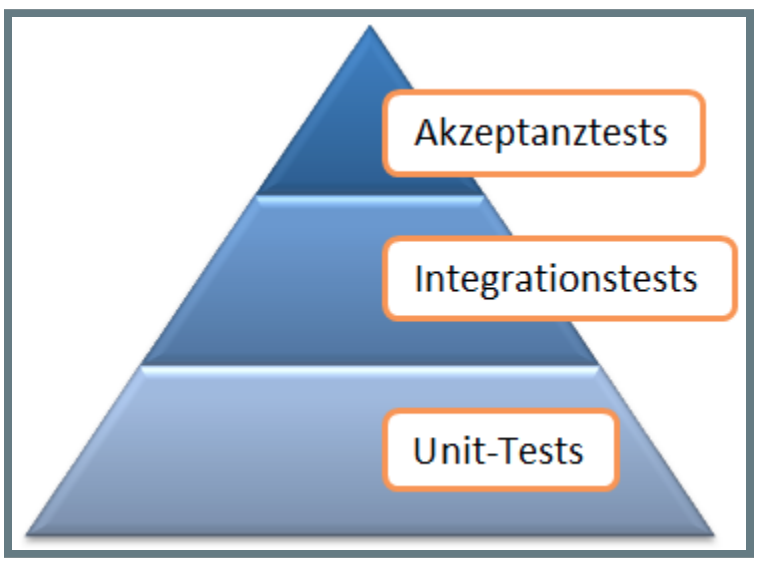
ENTWICKLUNGSPROZESS

AGILE - FEATURE BRANCHES / PULL REQUESTS - CODE REVIEWS - CLEAN CODE

- Master-, Develop-, Feature Branch*
- Pull Request = Feature → Develop
- Release = Develop → Master
- Keiner kann allein Code in das Repository bringen
- Features können unabhängig abgenommen werden
- Annahme von Pull Requests / Definition of Done
  - Testabdeckung
  - Code Review
  - Metriken (Komplexität, …)
  - Fachtest

TESTBARKEIT

- Test Driven Development (TDD)
- Stateless
- (Service)-Mocks
- Sehr dünner UI Layer (Oberflächentests sind teuer)
- Testdatenbereitstellung
- **Automatisierung**



BETRIEB

- Hohe Automatisierung
- Standardisierung auf Platform-Level (PAAS)
- Keine *Snowflake* Server
- Zentrales Monitoring
- Zentrales Logging

VIELEN DANK

FÜR IHRE AUFMERKSAMKEIT

BILD REFERENZEN 1/2

- Folie "Docker vs. VM", Urheber: Viktor Farcic, Quelle: The DevOps 2.0 Toolkit - Cloudbees, 2016
- Folie "start button", Quelle: pixabay.com, Link: https://pixabay.com/de/black-power-taste-ein-ausschalter-1428134/
- Folie "periodic table of devops tools", Link: https://xebialabs.com/periodic-table-of-devops-tools/
- Folie "Whitewater Championship", Urheber: Christopher J. Amelung, Quelle: Wikipedia, Link: https://de.wikipedia.org/wiki/Datei:St._Francis_River_C-1_Missouri_Whitewater_Championship_2008. jpg

BILD REFERENZEN 2/2

- Folie "Testbarkeit", Urheber: Johnny Graber, Link: https://graberj.wordpress.com/2013/03/10/die-testpyramide/