Continuous Software Engineering, DevOps und Docker

Spezielle Gebiete zum Software Engineering,
Nils Dralle

Continuous Software Engineering

- Vorgehensweise zur iterativen Softwareentwicklung
- Wird häufig in der agilen Softwareentwicklung eingesetzt

Continuous Integration

- Build-Automatisierung
- Unit-Test

Continuous Integration Build-Prozess

- Kompilieren
  - Sourcecode in ausführbares Format „übersetzen“
- Unit-Tests
  - Einfache Tests zum testen in sich geschlossener Einheiten der Anwendung
    - Junit
    - CxxTest
    - Jest

Continuous Integration
Build-Prozess

- Vorbereiten von Ressourcen
  - Dateien in ein anderes Format umwandeln, welches in der Produktionsumgebung besser geeignet ist
- Ausführbare Dateien
  - Beim kompilieren angefallene Artefakte „verpacken“
- Dokumentation
  - JavaDoc

Continuous Integration
Build-Tools

- Build-Prozess beschrieben durch ein Skript
- Unabhängig von Entwicklungsgerät

Continuous Integration Build-Tools

- Make

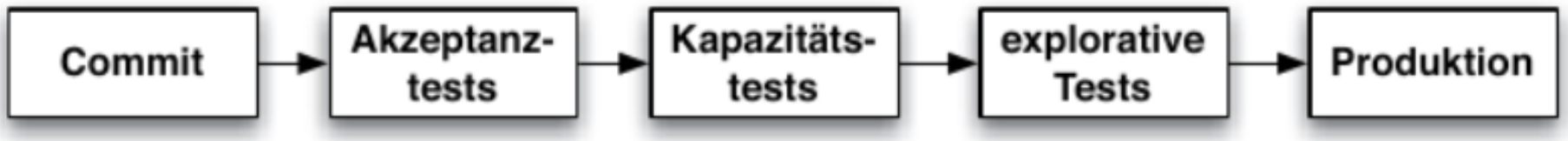
```
target: dependency
    gcc -Wall awesome.c -o awesomeExecutable
```

- Ant
  - Make in XML
- Maven
  - Ähnlich wie Ant in XML geschrieben
  - Geht von einer bestimmten Projektstruktur aus
- Gradle
  - Gradle-Skriptsprache
  - Geht wie Maven von einer bestimmten Projektstruktur aus

Continuous Delivery

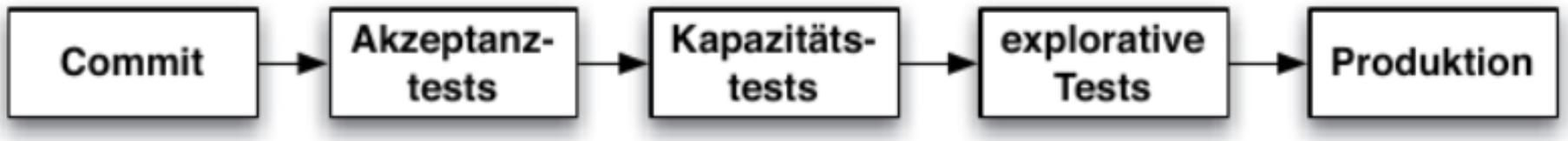
- Basiert auf Continuous Integration
- Automatisierung
- Reproduzierbarkeit

Continuous Delivery
Continuous Delivery Pipeline



- Commit
  - Continuous Integration
- Akzeptanztests
  - Automatisierte Tests zum Test der vom Kunden definierten Akzeptanzkriterien
- Kapazitätstests
  - Test der Anwendung unter Lastbedingungen
  - Skalierbarkeit
  - Nicht-funktionale Anforderungen

Continuous Delivery
Continuous Delivery Pipeline



- Explorative Tests
  - Automatisierte und manuelle Tests
  - Neuen Features, Verhalten
- Produktion
  - Installation der Anwendung

Continuous Testing

- Hauptsächlich manuelle Tests
  - Zum Beispiel explorative Tests
- Vom Entwicklerteam unabhängiges Testteam
- Test der neuesten Softwareversion

Continuous Deployment

- Ähnlich wie Continuous Delivery
- Nur automatisierte Tests
- Software wird per Definition an den Endkunden geliefert

DevOps

- IT-Development und IT-Operations
- Unterschiedliche Arbeitsweisen in Entwicklung und IT-Operations

DevOps
Agil

"Individuals and interactions over processes and tools"

- Mitarbeiter werden über die Prozesse gestellt
- Werkzeuge sollen die Arbeit unterstützen
- Verwendung geeigneter Werkzeuge

DevOps
Lean und Time to Market

- Was der Kunde nicht zahlt, ist verschwendet
- Fertige Arbeit soll möglichst schnell zu Geld gemacht werden

DevOps
Automatisierung

- Automatisierung einfacher und repetitiver Aufgaben
- Einsatz geeigneter Tools
- Mehrwert fürs Unternehmen

DevOps und Continuous Software Engineering

- Continuous Software Engineering ist eine Sammlung von „Best Practices“ zur Beschleunigung der Softwareentwicklung
- DevOps ist ein Werkzeug zur Prozessverbesserung in Unternehmen Techniken des Continuous Software Engineering werden bei DevOps eingesetzt.

Tools
Container

- Bündelt Anwendung mit Abhängigkeiten
- Reduziert Administrationsaufwand
- Virtualisierung oder Containerisierung

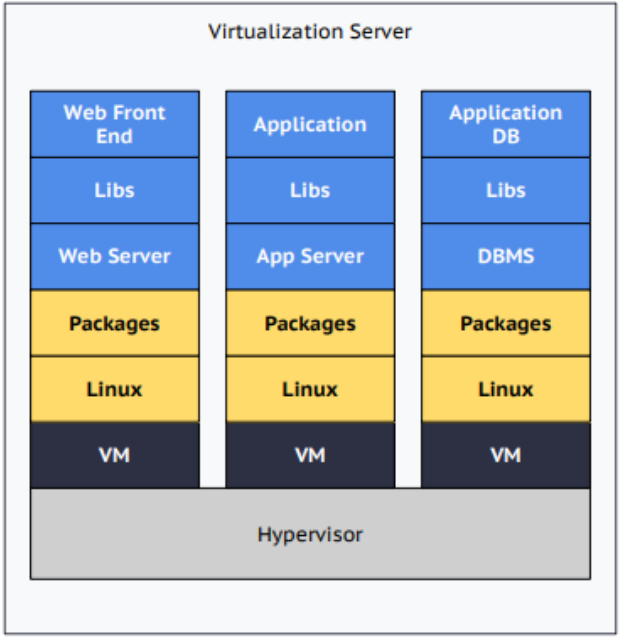
Tools
Jenkins

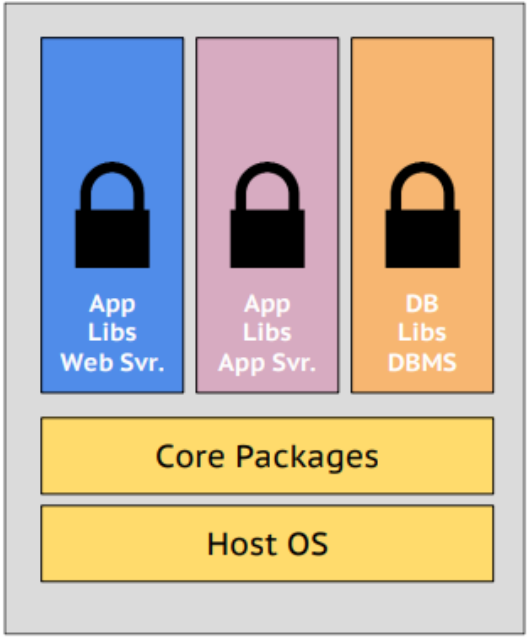
- Jenkins ist ein CI (Continuous Integration) Server
- Unabhängig von Programmiersprache
- Unabhängig vom verwendeten SCM
- Kann erweitert werden
- Unterstützt auch Continuous Delivery
  (mit Blue Ocean)

Tools
JIRA

- JIRA ist eine Plattform zum Projektmanagment
- Unterstützt agile Vorgehensmodelle
- Anforderungsmanagement, Statusmanagement, Fehlermanagement
- Alle Informationen sind als Tickets organisiert
  - Ein Ticket beschreibt eine Aufgabe

Tools
Container





Tools
Docker

- Anwendungs-Container
- Anwendung mit Abhängigkeiten
- Linux

Tools
Docker

- Linux
- Isolation vom Host-System
  - kernel namespaces
  - cgroups
  - containerd

Tools
Docker

- Nativ nur unter Linux
- Andere Betriebssystem → zusätzliche VM

Tools
Docker

- Image
  - Abbild der Software im Container
  - „Ausführbare“ Datei
- Layer
  - read-only
  - Set von Änderungen
  - Layer können „getaggt“ werden

```
cb09b21276d4: Pushed
24a0ccbb334b: Layer already exists
a93d5e407f48: Layer already exists
339fd2f0e879: Layer already exists
```

Tools
Docker

- Container
  - Führt Images aus
  - Layer können wie Images ausgeführt werden
- Dockerfile
  - Beschreibt, wie das Image gebaut wird

```
FROM 32bit/ubuntu:16.04
RUN apt-get update
RUN apt-get install -y openjdk-8-jdk-headless
VOLUME ["/var/hellojavadocker"]
COPY HelloDocker.jar /var/hellojavadocker/
CMD java -jar /var/hellojavadocker/HelloDocker.jar
```

Tools
Docker-Hub

- Repository für Docker-Images
- Integriert in Docker
- Viele fertige Images
  - Betriebssysteme
  - Serveranwendungen
  - ...
- docker push, docker pull

Ende

Vielen Dank für eure Aufmerksamkeit!

Ende

Fragen?