

Continuous Integration – stetige Kleinst-Änderungen

WEITER WISSEN →

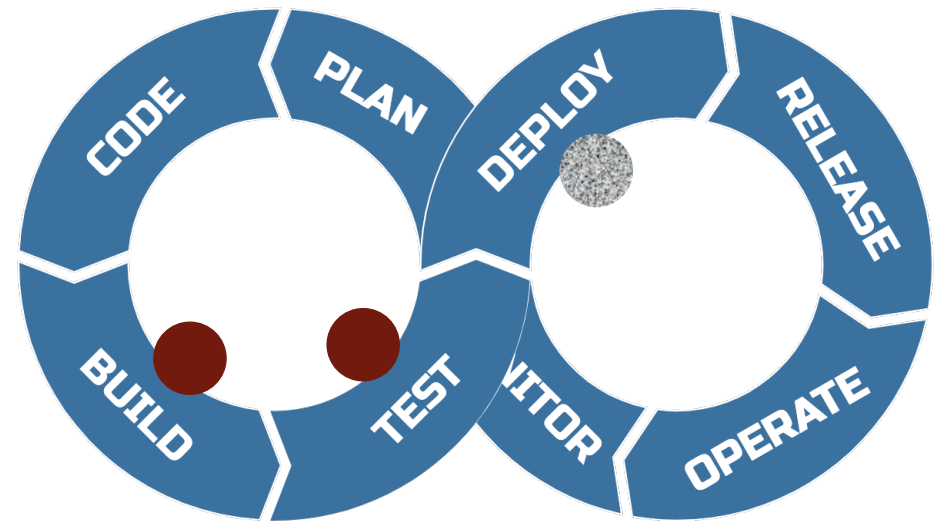


Ziel

Nach der Lektion legen die Studierenden dar, wie Änderungen kontinuierlich und in kleinen Stückchen in das Produkt eingepflegt werden und haben ein erstes Workflow File erstellt.

Continuous Integration

- Input: Docker Images (Wissen), Anwendungscode
 - später: Tests
- Output: Stets deploybares Artefakt für jeden Anwender
 - funktionsfähiges Docker Image
 - u.a. CCT
 - später in diesem Modul als letzter Teil
 - weitere Outputs sind z.B. Test Resultate, Helm Charts, Logs etc.
- Grundprinzip
 - Immer deploybar
 - «Trunk» ist nie «defekt»
 - Kleinste Änderungen
 - Keine BigBang Änderungen



Agenda

- Transferaufgabe 1
- Praktisches Beispiel
- Nutzwert
- Der erste eigene Workflow
- Vorgaben und Empfehlungen MA
- Zielkontrolle

Transferaufgabe 1

- Gute Discussions und Issues, sehr erfreulich!
- Aufgabe in Teams anschauen

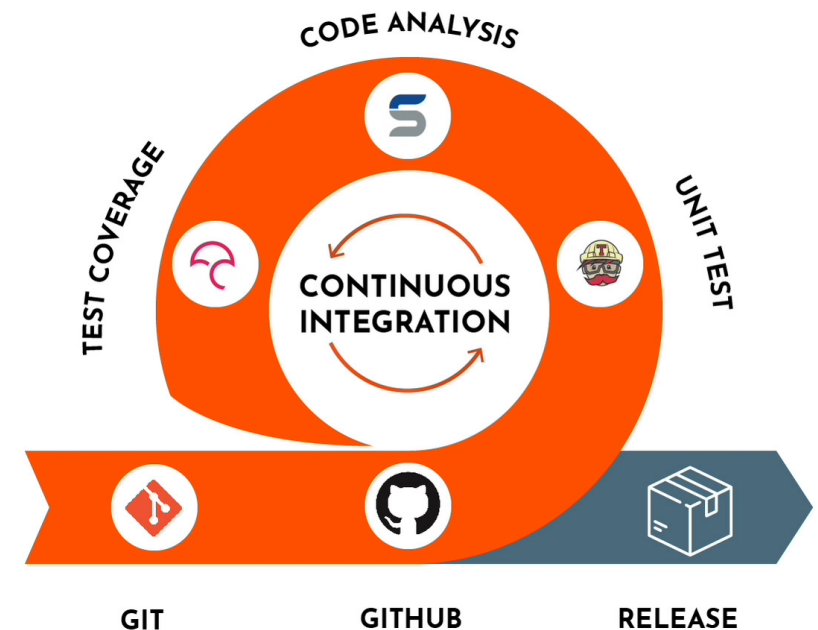
Vorgemacht - Praktisches Beispiel

- Kleine Änderung am ExMan Packer
 - Ich halte es für schlau, Java 8 benutzen zu wollen...

- Der Scan, der nun fehlschlägt, werden wir in Lektion 3 nachbauen!

Nutzwert

- Viele fancy Grafiken, schlussendlich
- Bestehende Codebase duplizieren
 - fork oder branch
- Kleine Änderung machen
- Pushen auf Branch
- CI läuft dagegen
 - Tests jeglicher Art inkl. Coverage
 - Statische Code Analysen
 - Lint
 - Licenses
 - Vulnerabilities
- Wenn OK, Merge
 - Artefakte bilden (jetzt erst)
- Sonst, nachbessern



CI in der (Modul-)Arbeit



GitHub Actions – CI der Zukunft (IMHO)

- Absolut überlegene, einfache Docs
- GitHub ist das (oder eines) der meistgenutzten VCS
- GitHub Actions sind direkt darin integriert
 - Gratis für Public / Open Source
 - Limitierte Minuten für Private (ausser bezahlt)
- Support von
 - Microsoft (nicht gerade klein)
 - Millionen «Contributors» und Firmen
- Eigene Actions

Es gibt auch andere gute CI Tools, aber GHA überzeugt mich mit Abstand am meisten!

Eigenes File erstellen

In einem Repository ihrer Wahl, erstellen Sie folgendes File

`.github / workflows / example-action_push.yml`

```
1  name: The most simple action ever
2
3  on:
4    push:
5      paths:
6        - .github/workflows/example-action_push.yml
7
8  jobs:
9    print-stuff:
10     runs-on: ubuntu-latest
11     steps:
12       - run: echo "Actions are amazing"
13       - run: echo "They run when you push and on many other occasions"
14       - run: echo "Learn more at https://docs.github.com/en/actions/quickstart"
```

Pushen Sie dieses in Ihr Repository, sehen Sie sowas?

<https://github.com/nds-swe/swdt/actions/runs/736246719>

Zusammenfassung

- Das ist die einfachste Action
 - Sie loggt einfach was
 - Sie löst aber nicht auf Pull Request aus sondern auf Push (und das auch nur bei dem bestimmten Pfad)
- Actions sind quasi gratis VMs zur Automation! 🤖💰
- In den nächsten Wochen werden wir darauf aufbauen
- Noch nie war CI so einfach, Actions sind erst paar Monate jung

Schaut Euch die besten Community Actions an:

<https://github.com/sdras/awesome-actions>

<https://docs.github.com/en/actions>

Vorgaben Modularbeit

9. **Richten Sie folgende Automatismen ein:**

- i. Ihr Code wird bei jedem Push zu einem Docker Image verpackt
- ii. Ihr Image wird bei jedem Push auf Vulnerabilitäten gescannt¹

4. **Richten Sie folgende Automatismen ein:**

- i. Die Unit Tests werden bei jedem Push ausgeführt
- ii. Die Integrationstests werden bei jedem Push ausgeführt
- iii. Der aus SWD gebildete Docker Container wird bei jedem Push gestartet und es wird überprüft, dass dieser erfolgreich «Healthy» wird¹

- Wir haben noch Wochen Zeit, das zu lernen
- Basis ist dieser super simple Workflow von eben
- Sie müssen sich sicher noch folgende Bereiche anschauen und aneignen
 - [actions/quickstart](#)
 - [reference/events-that-trigger-workflows#pull_request](#)
 - [reference/workflow-syntax-for-github-actions](#)
- Sie finden in meinen Repos stets tiefere Beispiele (lead by example)
 - z.B. den Docker Scan von Anfang Lektion im exman-packer

Zielkontrolle

Was machen diese Zeilen?

```
- run: echo "Actions are amazing"  
- run: echo "They run when you push and on many other occasions"  
- run: echo "Learn more at https://docs.github.com/en/actions/quickstart"
```

Was könnten wir sonst noch so einbauen?

Warum integrieren wir «kontinuierlich»?





WEITER WISSEN.

Wir begleiten Sie!