Don't just watch the Containers pass by:

How we (plan to) use Docker to Streamline the Kieker Development Process and Infrastructure

Thomas F. Düllmann
Symposium on Software Performance 2017
Karlsruhe, 2017-11-09

Why we are not done yet..

• Every stage runs in a clean Docker container
• Previous build results are copied over
• One task per stage
Why we are not done yet..

From Louda from Snap CI <lbaca@thoughtworks.com>
Subject Snap CI is going away
Reply to support@snap-ci.com
To ci@kieker-monitoring.net

Hello All,

We have some bad news for you today -- we are canceling Snap CI.

Feb 6th  Late Feb  Early May  Early July  Aug 1st
Continuous Delivery Pipeline

Stages

Checkout
• Git Checkout

Compile
• Compile Sources
• Compile Tests

Tests/Checks
• Junit Tests
• Static Analysis

Tasks

<table>
<thead>
<tr>
<th>Stages</th>
<th>Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Checkout</td>
<td>• Git Checkout</td>
</tr>
<tr>
<td>Compile</td>
<td>• Compile Sources</td>
</tr>
<tr>
<td></td>
<td>• Compile Tests</td>
</tr>
<tr>
<td>Tests/Checks</td>
<td>• Junit Tests</td>
</tr>
<tr>
<td></td>
<td>• Static Analysis</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Checkout</th>
<th>1-compile logs</th>
<th>2-unit-test logs</th>
<th>3-static-analysis logs</th>
<th>4-release-check-short logs</th>
<th>5-release-check-extended logs</th>
<th>push-to-stable</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1min 47s</td>
<td>51s</td>
<td>4min 32s</td>
<td>3min 18s</td>
<td>3min 38s</td>
<td>9min 22s</td>
</tr>
<tr>
<td></td>
<td>58s</td>
<td>53s</td>
<td>4min 35s</td>
<td>3min 12s</td>
<td>3min 19s</td>
<td>10min 34s</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>53s</td>
</tr>
</tbody>
</table>

How we (plan to) use Docker to Streamline the Kieker Development Process and Infrastructure
Thomas F. Düllmann, University of Stuttgart
What to do next?

• Keep the ship afloat

• Find an alternative for SnapCI

• Improve infrastructure and processes based on our experiences

What we came up with

• Jenkins with pipeline plugin (decision based on a student project)
Why (even more) Docker?

- **General**
  - Fixed state/setup
  - Independent from OS, IDE, configuration

- **Users**
  - Provide prepared setups with low effort

- **Developers**
  - Predefined/unified environment → reproducibility
What we did: Kieker LiveDemo

Possible improvements:

- automate Ansible runs
- Tests for LiveDemo

- docker run -ti -p 80:8080 kieker/livedemo:release
- Open [http://localhost/livedemo](http://localhost/livedemo)
- [http://demo.kieker-monitoring.net/livedemo](http://demo.kieker-monitoring.net/livedemo)
What we did: Kieker Building Process

GitHub → docker hub

• Possible improvements:
  • replace automated build
  • automate Ansible runs
Lessons learned

• If you put things in a container, you need a way to get them out again

• Easier and more consistent configuration rollouts with automation tools

What we achieved already
What we plan to do with Docker

• Integration tests

• Getting started with Kieker

• Docker local build tooling to provide more insight
Discussion

• Further ideas to use Docker in an open-source project

• Other technologies that could be helpful