# Testcontainers Integration in Build Pipelines

|  |  |
| --- | --- |
| **Fragestellung** | **Wie sollen Sidecars für die Unterstützung von Testcontainers in den Build Pipelines integriert werden?**  Einer erster PoC im Rahmen von ESTA-5359 hat gezeigt, dass mit Kubdock ein Sidecar für Build Tasks gestartet werden kann, welcher das notwendige Docker API für Testcontainers zur Verfügung stellt. Die Frage ist nun, wie dieser Sidecar in den ESTA Tekton Pipelines konfiguriert werden soll. Sidecars lassen sich nicht via Task Parameter ein/ausschalten sondern sind fix Bestandteil der Task Definition.  [ESTA-5359](https://flow.sbb.ch/browse/ESTA-5359) - ESTA Tekton: Unterstützung testcontainer Closed |
| **Rahmenbedingung** | * Tekton läuft auf Openshift auf AWS Clustern * Build Stacks welche Testcontainers unterstützen: Java, Node, Go, Python |
| **Annahmen** | * Sidecars lassen sich nicht via Task Parameter ein/ausschalten * kubedock Image ist 72.4MB gross * kubedock server hat kaum messbaren Ressourcen-Verbrauch (Memory/CPU) |
| **Alternativen / Varianten** | Variante 1 **Kubedock Sidecar ist Teil der Default Build Tasks und wird immer gestartet.**  Alle Build Tasks für Stacks, welche Testcontainers unterstützen, werden fix mit einem Kubedock Sidecar erweitert. Dieser startet immer einen zusätzlichen Container, ungeachtet dessen, ob für den Build Testcontainers verwendet werden.   |  |  | | --- | --- | | **Vorteile** | **Nachteile** | | Einfache Integration | Ressourcen Overhead | | Keine spezifischen Konfiguration nötig | Wird in 9x% aller Builds nicht verwendet | | Keine Erweiterung an estaTektonPipeline.json nötig |  |  Variante 2 **Je nach Pipeline Parameter werden spezifische Build Tasks mit Kubedock Sidecar referenziert.**  Für Stacks, welche Testcontainers unterstützen, werden spezifischen Build Tasks in den Pipeline Templates mit einem Kubedock Sidecar geführt. Je nach Pipeline Parameter werden dann diese Tasks anstelle der normalen esta-\*-build Tasks eingebunden.   |  |  | | --- | --- | | **Vorteile** | **Nachteile** | | Kein unnötiger Ressourcenverbrauch | Duplicated Code in den Task Templates | | Explizite Aktivierung im Repo (via estaTektonPipeline.json ) | Erweiterung von estaTektonPipeline.json nötig | |  | Spezifischen Konfiguration nötig |  Variante 3 **Kubedock Server direkt in Build Task starten.**  Das Kubedock Binary wird bei Bedarf (z.B. Pipeline Parameter) direkt im entsprechenden Build Task gestartet und exposed das Docker API lokal im Container. Es wird also kein Sidecar benötigt, dafür muss das Binary in den entsprechenden Builder Images der unterstützten Build Stack integriert werden. Wichtig hier ist, dass am Ende des Tasks der Kubedock Server wieder sauber heruntergefahren wird, also auch im Fehlerfall, wo das Task Script normalerweise abbricht (set -e).   |  |  | | --- | --- | | **Vorteile** | **Nachteile** | | Kein Sidecar notwendig | Builder Images +30MB grösser | | Explizite Aktivierung im Repo (via estaTektonPipeline.json) | Shutdown des Kubedock Servers muss sichergestellt werden | |
| **Entscheidung** | **Variante 3** |
| **Begründung** | Direktes Einbinden von Utilities in die Builder Image entspricht dem Pattern, was wir aktuell in ESTA Tekton verfolgen. Impact bezüglich Overhead ist in dieser Variante am kleinsten. |
| **Wer** | [Spirig Lukas (IT-PTR-CEN2-SL3)](file:///C:\display\~u228304) [Wallrapp Manuel (IT-PTR-EXT-EXT2 - Extern)](file:///C:\display\~ue64007) [Brüderli Thomas (IT-PTR-CEN2-SL2)](file:///C:\display\~u233658) |
| **Wann** | 22.06.2023 |