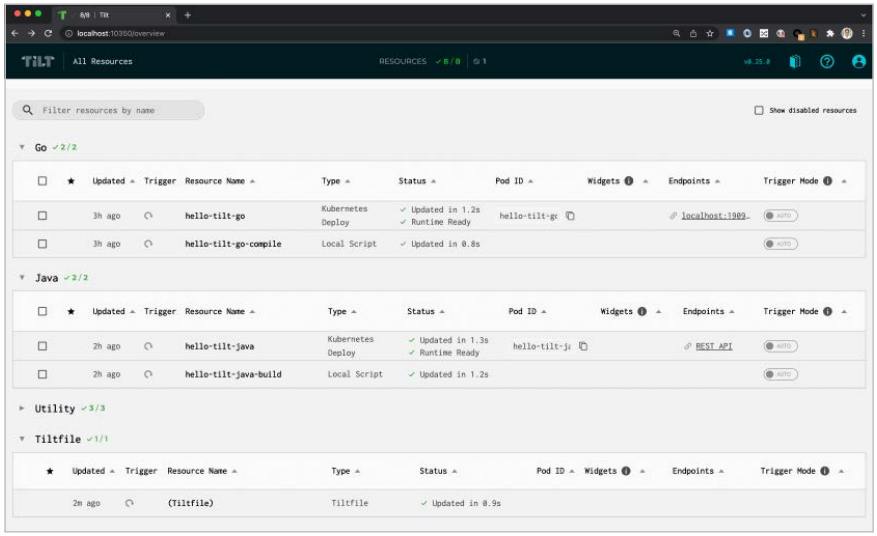

Microservices effizient mit Tilt entwickeln

Tilt up!

Mario-Leander Reimer

Das Open-Source-Tool Tilt stellt eine Entwicklungsumgebung für das Kubernetes-Deployment bereit. Es automatisiert die einzelnen Schritte der Microservices-Entwicklung und macht sie effizienter.

Das Kommandozeilenwerkzeug Tilt ähnelt in seiner Kernfunktion Skaffold [1]. Beide Tools übernehmen die nötigen Schritte bei der Entwicklung Cloud-nativer Anwendungen und Microservices vom Quellcode bis zur lokal lauffähigen Anwendung. Tilt verfolgt jedoch einen anderen Ansatz: Statt die Schritte deklarativ per YAML-Syntax zu beschreiben, verwendet es die imperative Sprache Starlark, einen speziellen Python-Dialekt mit Schwerpunkt auf Konfigurationsaufgaben. Die Definition der Schritte ist dadurch kompakter und auch komplexe Abläufe lassen sich einfach per Code umsetzen.

Das Installieren auf dem Entwickler-PC ist schnell erledigt. Tilt stellt für alle gängigen Plattformen und Betriebssysteme ein Shellskript für die Installation bereit. Auch das manuelle Installieren per Download des Binary von der GitHub-Releaseseite des Projektes ist möglich. Zusätzlich ist Tilt auch in Paketmanagern wie Homebrew, Scoop oder asdf verfügbar. Lediglich eine lokale Docker-Installation, das kubectl-Tool und ein erreichbarer Kubernetes-Cluster sind Voraussetzung.

Um einen ersten Eindruck von der Funktionsweise von Tilt zu bekommen, nutzt man den Befehl `tilt demo`, der einen temporären lokalen Kubernetes-Cluster erzeugt, den Check-out eines Beispielprojektes ausführt und anschließend die kontinuierliche Build- und Deployment-Schleife über das Kommando `tilt up` initiiert.

Die Definition der Build- und Deployment-Schritte erfolgt in der Datei Tiltfile im Root-Verzeichnis des Projektes. Tiltfile ist ein Starlark-Programm, das über gängige Programmierkonstrukte wie Schleifen, Variablen und Funktionen verfügt (Beispiel siehe ix.de/z2hq). Tilt bringt zudem 75 spezielle Funktionen mit.

Ressourcen sind wichtiges Element

Das zentrale Konzept in Tilt ist die Ressource. Sie ist eine Menge an Arbeitsschritten und überwachten Artefakten, die Tilt verwaltet. Die Beziehung und korrekte Reihenfolge der Ressourcen erkennt Tilt in den meisten Fällen automatisch. Die Abhängigkeiten lassen sich aber auch explizit definieren und damit einschränken. Notwendige Ressourcen sind neben der automatischen Ausführung auch manuell aktivierbar.

Mit dem Befehl `local_resource` können Entwickler beliebige lokale Befehle ausführen und das lokale Verzeichnissystem auf Änderungen überwachen. Dieses flexible Konstrukt erlaubt, benötigte Arbeitsschritte und CLI-Kommandos einfach zu automatisieren, ganz gleich, ob es sich um ein Java-Projekt mit Gradle Build oder ein Go-Projekt mit Makefile handelt. Zum Bauen von Container-Images stehen neben dem einfachen `docker_build` die `custom_build`-Ressourcen zur Verfügung, über die man andere Tools wie Jib, Bazel und ko (für Go) sowie kaniko oder Google Cloud Build in die Tilt-Pipeline einbinden kann. Zum Rendern der Kubernetes-Ressourcendefinitionen unterstützt Tilt Plain-YAML-Dateien, Kustomize-Overlays und Helm. Reicht der Funktionsumfang von Tilt nicht aus, kann er durch Extensions und das Nachladen neuer Funktionen erweitert werden.

Highlights von Tilt sind die textbasierte Konsole und das webbasierte Dashboard, das nach dem Starten der Pipeline mit `tilt up` verfügbar ist. Beide zeigen den Status und weitere Details zu den verwalteten Ressourcen. Zur besseren Übersichtlichkeit lassen sich die Ressourcen sortieren, gruppieren und filtern. Die Logs aller Container werden gesammelt und auf der Oberfläche ausgegeben. Für die exponierten Ports können Entwickler zusätzliche Links zum Schnellzugriff konfigurieren und darstellen. Die detaillierte Statusanzeige vereinfacht das Troubleshooting im Fehlerfall erheblich.

Tilt ist Open Source und steht unter der Apache License 2.0. Es wurde von Windmill Engineering entwickelt. (nb@ix.de)

Quellen

[1] Mario-Leander Reimer; Wie vom Fließband; Kontinuierliche Anwendungsentwicklung mit Skaffold; *iX* 2/2022, S. 139

[2] Beispielprojekt, Tilt-Extensions, Release- und Projektseite: www.ix.de/z2hq

Mario-Leander Reimer

ist Principal Software Architect bei der QAware GmbH. Er beschäftigt sich mit Innovationen und Technologien rund um den Cloud-native-Stack.