



## Ab in die Cloud-Gegenwart!

Java-Software-Artefakte waren schon immer – by Design – recht portabel, die zugehörige Infrastruktur indes nicht. So blieb zwar das stets proklamierte „Write once, run everywhere“ in der Java-Community als Phrase lebendig, in der Praxis hatte es wenig Bedeutung. Mit Docker entstand im Jahre 2014 die Möglichkeit, Software als Ganzes inklusive der Ablaufumgebung in Container verpackt und mit Hilfe der Docker Registry bereitzustellen.

Später setzte sich ergänzend Kubernetes durch, um die neue, mit Docker containerisierte Cloud-native Microservices-Welt zu organisieren. Dass aber der Kubernetes-Standard allein auch noch keine Lösung darstellt, verdeutlicht ein Blick auf die Cloud-Native-Landkarte der Cloud Native Computing Foundation: <https://landscape.cncf.io>. Es bleibt kompliziert.

Um leistungsfähige Systeme robust, flexibel, skalierbar und kosteneffizient in der Cloud zu deployen und zu betreiben, bedarf es vieler Stellschrauben, die sorgfältig zu justieren sind. Das scheint dem Versprechen von DevOps, die Verwendung von Infrastruktur durch Automatisierung zu vereinfachen, Hohn zu sprechen. Das schöne neue „Infrastructure as Code“-Universum scheint, mal wieder, Java-Entwickler:innen von ihrer Kernaufgabe abzulenken, nämlich: guten Applikationscode zu schreiben, der den Businessanforderungen dient.

Hinzu gesellt sich ein weiteres Problem: Java wurde in seiner über 25-jährigen Geschichte nicht wirklich für diese Art von Anforderungen designt, wie Lars Röwekamp, Thilo Frotscher und Frederieke Scheper in ihren Leitartikeln erläutern.

Auftritt Quarkus: „A Kubernetes Native Java stack [...] crafted from the best of breed Java libraries and standards.“ Das klingt nach entschlossenem und ehrlichem Anpacken, was uns die Leute von Red Hat da auf der Quarkus-Website versprechen; ein Fullstack-Framework, das Java-Entwickler:innen einen Weg durch den Cloud-nativen Dschungel weisen und ihnen die Freude beim Entwickeln von Applikationen (und nicht der Infrastruktur) zurückgeben will.

Wenn diese Rechnung aufgeht – und unser Titelthema legt diesen Schluss nahe – dann hat die Java-Welt wieder einmal bewiesen, dass Totgesagte einfach länger leben, denn Quarkus hat das Zeug dazu, Java aus der alten Welt direkt in die Cloud-Gegenwart zu katapultieren!

Dass dies unter Vermeidung von Technologiebrüchen gelingen soll, das heißt durch Verwendung klassischer APIs und Libraries der Java-Welt, macht Quarkus zu einem Anwärter für die Hall of Fame der Java-Innovationen, denen es immer wieder gelungen ist, Innovation mit Investitionsschutz zu versöhnen.

Eine inspirierende Lektüre wünscht

**Sebastian Meyen** | Chefredakteur



@JavaMagazin