

# Infrastructure as Code

## Systembetrieb trifft Software-Engineering

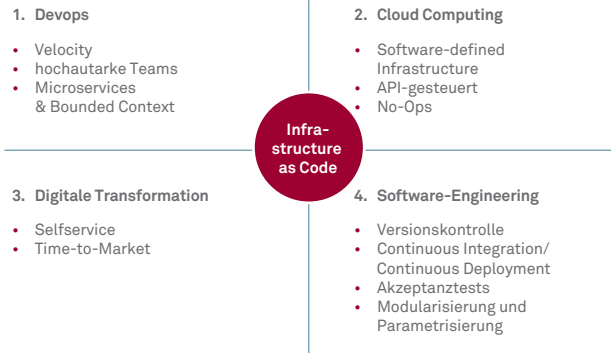
Infrastructure as Code ist die automatisierte und standardisierte Bereitstellung von mitunter komplexen IT-Infrastrukturen. Das Ergebnis dieser Automatisierung ist eine konsistente und wiederholbare Infrastruktur bei einer stark erhöhten Geschwindigkeit der Produktentwicklung.

### Definition

Infrastructure as Code (kurz IaC) ist ein Ansatz zur automatisierten Bereitstellung von Infrastruktur basierend auf etablierten Praktiken der Software-Entwicklung. Es betont konsistente, wiederholbare Abläufe zur Provisionierung und Konfiguration von IT-Infrastruktur, angefangen beim Speicher über das Netzwerk hin zur virtuellen Maschine, dem Betriebssystem und darauf laufender Middleware.

Infrastructure as Code betrachtet jedwede Infrastruktur, typischerweise in einer Cloud-Umgebung, als Software mit Daten und behandelt sie dementsprechend. IaC-Methoden bilden somit nicht nur den kompletten Software-Lebenszyklus ab. Sie wenden auch die aus dem Umfeld der Software-Entwicklung bekannten Techniken an, etwa Versionskontrolle, Testautomatisierung, Modularisierung, Parametrisierung sowie kontinuierliche Integration und Auslieferung.

Das Ergebnis ist die häufig deklarative Automatisierungsdefinition einer IT-Infrastruktur, ausgedrückt durch Quellcode. Entsprechende Tools parametrisieren den Code und wenden ihn auf

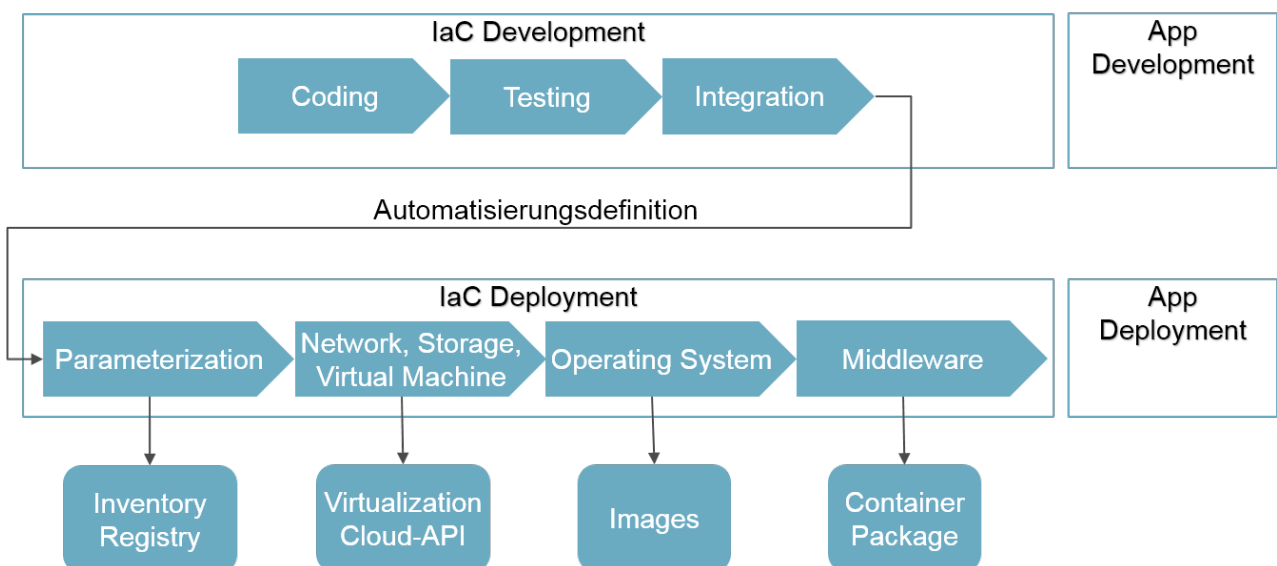


die zur Verfügung stehende Virtualisierung oder Cloud-API an. So setzen sie das als Quellcode definierte Zielbild der Systeme um.

Der Mehrwert von Infrastructure as Code lässt sich in drei messbare Bereiche herunterbrechen: gesteigerte Kosteneffizienz, Unterstützung von schnellen, agilen Entwicklungsteams und verringerte Betriebsrisiken.

### Referenzszenario

Eine typische Ausgangslage ist ein Produkt, dessen Entwicklung nach agilen Methoden und Devops-Ansätzen verläuft. In solchen



Fällen gibt es verschiedene fachliche Domänen mit Expertenteams, die für jeweils ein Teilprodukt die Verantwortung tragen und ihr Produkt als unabhängig auslieferbaren Microservice entwickeln.

Um dieser Architektur Herr zu werden, gibt es einen Bedarf nach komplexer und umfangreicher IT-Infrastruktur. Durch den Einsatz von Cloud-Technologien steht den Teams eine dynamische, unterliegende Infrastruktur zur Verfügung. In dieser Infrastruktur können Teams unabhängig vom Betrieb – und ohne sich gegenseitig zu verhindern – ihre Entwicklung vorantreiben.

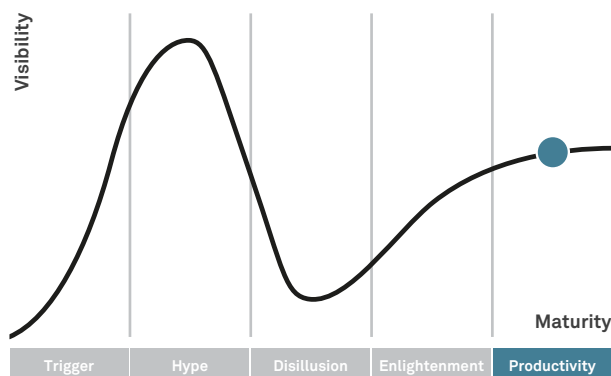
Infrastructure as Code bietet den Produktteams eine Möglichkeit, ihre benötigten Ressourcen automatisiert, konsistent und wiederholbar bereitzustellen. Darunter fallen selbstverständlich virtuelle Maschinen mit Datenbanken und Middleware, aber auch Elemente wie Cluster, Loadbalancer und Netzwerke. Abschließend lässt sich die erzeugte Infrastruktur wieder automatisiert und restlos abbauen.

### Potenzial

In Zeiten der Digitalen Transformation spielt eine effiziente und effektive IT eine entscheidende Rolle. Immer wichtiger werdende Faktoren, um sich maßgebliche Wettbewerbsvorteile gegenüber Mitbewerbern erarbeiten zu können, sind eine hohe Flexibilität und eine hohe Qualität der für das Unternehmen eminenten IT-Infrastruktur. Infrastructure as Code unterstützt dieses Ziel perfekt und baut auf bewährtem Software-Engineering auf.

### Reifegrad

Es gibt viele qualitativ hochwertige Tools, die eine große Akzeptanz auf dem Markt haben. Virtualisierung und Cloud-Umgebungen als Basis-Technologien für Infrastructure as Code sind mittlerweile gut verstanden. Software-Engineering selbst hat einen hohen Reifegrad sowie viele allgemein bekannte Best Practices und etablierte Methoden.



### Buzzword Factor (Ent./Customer)

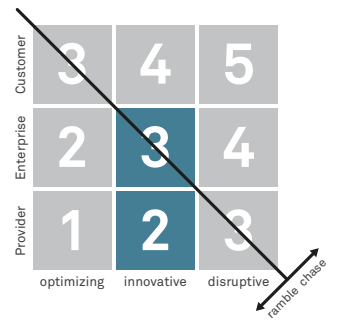
1 low	2 medium	3 high
----------	-------------	-----------

### Entry Barrier (Provider)

1 low	2 medium	3 high
----------	-------------	-----------

### Benefit Level (Provider)

1 low	2 medium	3 high
----------	-------------	-----------



### Marktübersicht

Die Tools für Infrastructure as Code sind vielfältig und decken unterschiedlichste Teilbereiche ab. Sie lassen sich grob in die fünf Kategorien Konfiguration, Ressourcenverwaltung, Cloud, Virtual-Machine-Monitoring (Hypervisor) und Clustering unterteilen.

Im Bereich der Konfiguration sind Ansible, Puppet, Chef, Saltstack und Vagrant prominente Vertreter, im Bereich Ressourcenverwaltung hingegen Cloudformation und Terraform. Amazon Web Services, Microsoft Azure und Google Cloud Platform decken den Bereich Cloud ab. Für virtuelle Maschinen und deren Überwachung stehen Virtualbox, Vmware und Microsoft Hyper-V zur Auswahl. Clustering ermöglichen Consul, Etcd, Kubernetes und Openshift.

### Alternativen

Eine Alternative für IaC kann der Einsatz einer Ad-hoc-Automatisierung ohne Software-Engineering sein. Im sehr kleinen Rahmen kann sich auch ein manueller Eingriff in die Infrastruktur anbieten. Schlussendlich ist es möglich, höhere Schichten wie SaaS- oder FaaS-Dienste zu nutzen und damit IaC dem Provider zu überlassen.

Pro	Contra
reproduzierbare Prozesse und Systemlandschaft	große Macht bringt große Verantwortung für die Anwender von IaC mit sich
eine Quelle der Wahrheit	Gefahr des Geschwindigkeitsverlusts, weil ein kleiner Fix einen komplexen Eingriff erfordern kann
hohe Effizienz und Geschwindigkeit durch Automatisierung	Fähigkeiten aus dem Bereich Software-Engineering notwendig
Reduzierung von Betriebsrisiken	Spirale der Automatisierungsangst muss durchbrochen werden