Container-Technologie

Ein umfassender Überblick: Wie Software durch Container-Lösungen schneller, einfacher und zuverlässiger entwickelt und deployed wird

Inhalt

Die Grundlagen

1. WAS IST CONTAINER-TECHNOLOGIE ODER CONTAINER-VIRTUALISIERUNG?

Ein Container ist eine standardisierte, in sich geschlossene Software-Einheit. Container-Virtualisierung ermöglicht es, eine Anwendung unabhängig vom Host-Betriebssystem (OS) zu betreiben. Der Container kapselt eine gesamte Anwendung – einschliesslich Code, Abhängigkeiten und Konfiguration – in einem klar definierten, portablen Format auf Basis des **Open Container Initiative (OCI)**-Standards.

Der OCI-Standard sorgt dafür, dass Images, die mit einem beliebigen OCI-kompatiblen Tool (Docker, Podman, Buildah, Kaniko usw.) erstellt wurden, ohne Anpassung auf jeder OCI-kompatiblen Container-Runtime und Orchestrierungsplattform, einschliesslich Kubernetes, laufen.

---

2. WAS SIND DIE VORTEILE EINER CONTAINER-LÖSUNG?

Die Standardisierung eines Containers und seine Unabhängigkeit vom Host-OS bieten attraktive Vorteile für die Entwicklung und den Betrieb von Software:

- **Anwendungen lassen sich einfacher skalieren:** Container-Virtualisierung basiert auf autonomen, unabhängig funktionierenden Einheiten. Anwendungen können schnell und flexibel nach Bedarf hoch- oder herunterskaliert werden.
- **Container starten schneller:** Kein dediziertes Gast-Betriebssystem muss hochgefahren werden: Startzeiten werden in Millisekunden gemessen.
- **Container nutzen Ressourcen effizient:** Ohne den Overhead eines dedizierten OS pro Anwendung sparen Container aktiv physische Ressourcen, die anderen Workloads zur Verfügung stehen.
- **Agilere Softwareentwicklung:** Container sind schnell zu erstellen und zu deployen, und sie sind reproduzierbar: Ein Container in der Entwicklungsumgebung ist identisch mit dem in der Produktion.
- **Container sind portabel:** Container-Anwendungen laufen weitgehend unabhängig vom Host-OS und können mit minimalem Aufwand zwischen Infrastruktur-Anbietern migriert werden.

3. WAS SIND DIE UNTERSCHIEDE ZWISCHEN CONTAINER-LÖSUNGEN UND VIRTUELLEN MASCHINEN (VMS)?

Container und virtuelle Maschinen repräsentieren zwei verschiedene Ansätze zum Hosting und Betrieb von Anwendungen in einer virtuellen Umgebung. Bei VMs verteilt ein Hypervisor die physischen Serverressourcen auf virtuelle Maschinen, die jeweils ein dediziertes Betriebssystem benötigen. Container hingegen teilen sich den Host-OS-Kernel: Virtualisierung erfolgt auf Betriebssystemebene. Daher nutzen Container vorhandene Ressourcen deutlich effizienter als VMs und starten schneller.

Container und VMs schliessen sich nicht gegenseitig aus: Die meisten Kubernetes-Produktionsumgebungen betreiben containerisierte Workloads auf VM-Nodes und kombinieren so die Isolationsvorteile von VMs mit der betrieblichen Effizienz von Containern.

---

4. WELCHE RISIKEN SIND MIT EINER CONTAINER-LÖSUNG VERBUNDEN?

Wer sich intensiver mit Container-Technologie beschäftigt hat, ist möglicherweise auf folgende Bedenken gestossen:

> **«Container sind für unsere Anforderungen nicht sicher genug.»**
> Container sind nicht weniger sicher als andere Umgebungen. Die Sicherheit hängt primär davon ab, wie die Umgebung konfiguriert und gewartet wird. Korrekte Pod Security Standards, Image-Scanning, Network Policies und Secret Management sind entscheidende Faktoren. Eine gut konfigurierte Container-Umgebung ist oft auditierbarer und konsistenter als traditionelle Deployments.

> **«Container-Management ist viel zu komplex für uns.»**
> Wie bei jeder Veränderung ist die Migration zu einem containerbasierten System mit einem initialen Aufwand verbunden. Anwendungen müssen vorbereitet, Teams geschult und Deployment-Prozesse automatisiert werden. Diese Investition zahlt sich jedoch schnell aus. Die Komplexität lässt sich durch eine Managed-Kubernetes-Plattform erheblich reduzieren, sodass dein Team sich auf die Anwendungsentwicklung konzentrieren kann.

> **«Wir haben bereits eine Deployment-Strategie. Wir brauchen keine Container.»**
> Container ersetzen keine Deployment-Strategie, sondern ergänzen und optimieren sie. Es gibt verschiedene Möglichkeiten, Container in eine bestehende IT-Strategie zu integrieren, auch ergänzend zu vorhandener VM-basierter Infrastruktur.

Die Implementierung

5. WELCHE KONKRETEN VERBESSERUNGEN BRINGT CONTAINER-TECHNOLOGIE BEI UPDATE- UND DEPLOYMENT-PROZESSEN?

Bei Update- und Deployment-Prozessen sind Geschwindigkeit und Zuverlässigkeit die entscheidenden Grössen. Genau hier entfalten Container ihre Stärken:

- **Schnellerer Start:** Container starten in Millisekunden und ermöglichen schnelle Skalierung und Deployments
- **Kürzere Feedbackzyklen:** Entwickler erhalten durch optimierte CI-Pipelines schneller Testergebnisse
- **Einfachere Skalierung:** Gewünschte Replikaanzahl oder Autoscaling-Policy festlegen, die Plattform übernimmt den Rest
- **Rolling Deployments ohne Unterbrechung:** Neue Versionen werden sofort verfügbar, ohne Wartungsfenster
- **Identische Umgebungen:** Der Container in der CI-Pipeline ist identisch mit dem in der Produktion

---

6. WAS SIND CONTAINER-ORCHESTRIERUNGSPLATTFORMEN?

Container-Orchestrierungsplattformen verwalten einzelne Container im grossen Massstab. Sie bilden ein dynamisches System, das Container gruppiert und platziert, ihre Netzwerkverbindungen verwaltet und die Kommunikation zwischen ihnen sicherstellt. Diese Plattformen können On-Premise oder auf Cloud-Plattformen betrieben werden. Die am weitesten verbreitete Orchestrierungsplattform ist Kubernetes.

---

7. WAS IST KUBERNETES?

Kubernetes ist eine Open-Source-Plattform für die Orchestrierung containerisierter Workloads und Services. Es automatisiert das Deployment, den Betrieb, die Skalierung und das Management containerisierter Anwendungen. Laut dem CNCF Annual Survey 2024 nutzen **80 % der Unternehmen Kubernetes bereits in der Produktion**: Es ist der etablierte Standard für den Betrieb von Containern im grossen Massstab.

Kubernetes befindet sich derzeit auf Version **v1.33** (April 2025) und veröffentlicht drei neue Minor-Versionen pro Jahr, was das hohe Entwicklungstempo im Ökosystem widerspiegelt.

---

8. WAS IST DEVOPS?

DevOps als organisatorische und Führungskultur basiert auf drei Prinzipien («The Three Ways»): dem Flussprinzip, dem Feedbackprinzip und dem Prinzip des kontinuierlichen Lernens und Experimentierens. Diese dienen als Rahmen für Prozesse, Verfahren und Praktiken, die auf die Optimierung der Zusammenarbeit zwischen Entwicklungs- und Betriebsteams ausgerichtet sind, mit dem übergeordneten Ziel, Software schneller und zuverlässiger zu liefern.

---

9. SIND DEVOPS UND CONTAINER IMMER UNTRENNBAR?

DevOps als Philosophie und Container als Technologie ergänzen sich hervorragend, sind aber nicht voneinander abhängig. DevOps-Methoden können ohne Container angewendet werden, und Containerisierung ist ohne DevOps-Organisation möglich. Containerbasierte Workflows fördern jedoch von Natur aus Automatisierung, Reproduzierbarkeit und kurze Feedbackzyklen, alles zentrale Aspekte der DevOps-Arbeitsweise.

---

10. GIBT ES BEST PRACTICES FÜR DAS DEPLOYMENT VON CONTAINER-UMGEBUNGEN?

Helm Charts sind der De-facto-Standard für die Paketierung von Kubernetes-Anwendungen. Sie vereinfachen das Deployment komplexer Multi-Komponenten-Anwendungen erheblich. Starte mit Helm: Die meiste Software von Drittanbietern wird als Helm Chart ausgeliefert.

Für das Kubernetes-Konfigurationsmanagement bietet **Kustomize** (in kubectl integriert) einen einfachen, template-freien Ansatz für umgebungsspezifische Konfigurationsüberlagerungen. Für Teams, die lieber in Code schreiben, bieten Tools wie **cdk8s** oder **Pulumi** programmierbarere Ansätze.

Für Produktions-Deployments gilt das **GitOps**-Muster (mit Tools wie ArgoCD oder Flux) inzwischen als Best Practice.

---

11. WIE STEIL IST DIE LERNKURVE BEIM ARBEITEN MIT CONTAINERN?

Das hängt vom gewünschten Autonomiegrad ab. Wenn die Container-Plattform vollständig von einem externen Anbieter verwaltet wird, reichen grundlegende Container-Kenntnisse. Bei der Verwaltung der Plattform im eigenen Haus erfordert die Migration zu einer containerbasierten Architektur umfassende Schulungen für Entwickler und Systems Engineers. Ohne solides Fundament ist das Selbst-Management eines Kubernetes-Clusters in der Produktion riskant.

---

12. WAS IST DER BESTE EINSTIEG IN EIN CONTAINER-PROJEKT?

Ein praxisnahes Training zu Docker und Kubernetes bietet die ideale Grundlage. Wir bieten kostenlose Kubernetes-Webinare (eine Stunde, Grundlagen und NKE in der Praxis) sowie die ganztägige Nine Kubernetes Academy (CHF 800 / CHF 600 für NKE-Kunden) für Teams, die tieferes, praktisches Wissen erwerben möchten. Beide Formate wurden von unseren CKA-zertifizierten Engineers entwickelt und decken alles von der Cluster-Architektur bis hin zu produktionsreifen Deployment-Mustern ab.

---

13. CONTAINER-TECHNOLOGIE: MANAGED ODER UNMANAGED?

Aufgrund der Komplexität beim Aufbau und Betrieb der zugrundeliegenden Infrastruktur – und des schnellen Entwicklungstempos im Kubernetes-Ökosystem (drei Minor-Releases pro Jahr, jeweils mit Security-Updates und Breaking Changes) – empfehlen wir dringend, die Plattformverwaltung an einen Managed-Service-Anbieter zu übergeben. So kann sich dein Team vollständig auf die Anwendungsentwicklung konzentrieren, ohne den betrieblichen Aufwand für Cluster-Upgrades, Zertifikatsmanagement, Monitoring und Backup zu tragen.

---

14. WAS IST GITOPS UND WARUM IST ES WICHTIG?

GitOps ist ein Deployment-Muster, bei dem der gewünschte Zustand eines Kubernetes-Clusters in einem Git-Repository deklariert wird und ein automatisierter Agent den Live-Cluster-Zustand kontinuierlich mit dem in Git deklarierten Zustand abgleicht. Tools wie **ArgoCD** und Flux implementieren dieses Muster. Die Vorteile: vollständig auditierbare Deployment-Historie, einfache Rollbacks, automatische Drift-Erkennung und eine klare Trennung zwischen Infrastrukturzustand und Anwendungscode. GitOps gilt heute als Produktions-Best-Practice für Kubernetes-Deployments.

Sicherheit und Netzwerk

15. WELCHE NETZWERKSYSTEME UND -ARCHITEKTUREN SIND MIT CONTAINERN KOMPATIBEL?

Container können als sinnvolle Ergänzung oder Alternative zu VMs eingesetzt werden. Sie können bei Bedarf auch mit VM-basierten Systemen innerhalb eines Netzwerks verbunden werden. Kubernetes-Networking wird durch CNI-Plugins (Container Network Interface) gehandhabt: Beliebte Optionen sind Calico, Cilium und Flannel.

Traditionell nutzte Kubernetes Ingress-Ressourcen und -Controller (wie ingress-nginx) für das externe Traffic-Routing. Die Community standardisiert nun auf die **Gateway API**, die 2024 die Version 1.0 General Availability erreicht hat und einen ausdrucksstärkeren, rollenorientierten Ansatz für das Traffic-Management bietet. Neue Deployments sollten die Gateway API gegenüber Ingress für langfristige Kompatibilität evaluieren.

---

16. WIE SICHER SIND CONTAINER-LÖSUNGEN IM VERGLEICH ZU VIRTUELLEN MASCHINEN?

Container sind genauso sicher wie traditionelle Server-Lösungen und VMs: Entscheidend ist, wie die Umgebung konfiguriert und gewartet wird. Wichtige Sicherheitsaspekte in Kubernetes-Umgebungen:

- **Pod Security Standards (PSS):** Kubernetes v1.25 hat das veraltete PodSecurityPolicy (PSP) entfernt. Der aktuelle Mechanismus ist **Pod Security Admission (PSA)**, das die Sicherheitsprofile Privileged, Baseline oder Restricted durchsetzt. Zusätzliche Policy-Engines wie **Kyverno** oder **OPA Gatekeeper** bieten feingranularere Kontrolle.
- **Image-Scanning:** Container-Images vor der Produktion auf bekannte Sicherheitslücken (CVEs) scannen. Tools wie Trivy oder Grype automatisieren dies.
- **Secret Management:** Kubernetes Secrets sind standardmässig nur base64-kodiert. Für die Produktion externes Secret Management via External Secrets Operator, HashiCorp Vault oder einem Cloud-Provider-Secret-Store verwenden.
- **Network Policies:** Pod-zu-Pod-Kommunikation mit Kubernetes NetworkPolicy-Ressourcen einschränken, um ein Zero-Trust-Netzwerkmodell umzusetzen.

- **RBAC:** Role-Based Access Control limitiert, was Benutzer und Service Accounts im Cluster tun können.

---

17. GIBT ES ALTERNATIVEN ZU DOCKER FÜR DAS ERSTELLEN UND BETREIBEN VON CONTAINERN?

Das Container-Tooling-Ökosystem hat sich erheblich diversifiziert:

- **Für das Erstellen von Images: Podman** (daemonlos, rootless, Drop-in-Docker-CLI-Ersatz), **Buildah** (feingranulares OCI-Image-Building), **Kaniko** (Builds innerhalb von Kubernetes ohne Docker-Daemon)
- **Für Container-Runtimes in Kubernetes:** Seit Kubernetes v1.24 wurde die Docker-Runtime-Schicht (dockershim) entfernt. Die Standard-Runtime ist nun **containerd**, die über 95 % der Kubernetes-Cluster antreibt. CRI-O ist eine weitere OCI-konforme Option, die häufig mit OpenShift verwendet wird. Bestehende Docker-Images laufen ohne Änderungen auf containerd, da beide das OCI-Image-Format verwenden.
- **Wichtiger Hinweis:** Docker Desktop für kommerzielle Nutzung erfordert seit 2022 ein kostenpflichtiges Abonnement. Teams sollten Alternativen wie Podman Desktop, Rancher Desktop oder OrbStack evaluieren.

Das Team überzeugen

18. WIE KANN INTERNE AKZEPTANZ FÜR DIE EINFÜHRUNG VON CONTAINERN GESCHAFFEN WERDEN?

Wie bei jeder anderen Innovation sind konkrete Kosteneinsparungen und eine verbesserte Time-to-Market die überzeugendsten Argumente. Praktische Case Studies und Beispiele aus der Praxis sind äusserst wirksam. Zahlen und Geschichten zusammen machen den Fall. Die Erfahrungsberichte unserer Kunden mit dem Wechsel zu NKE sind auf nine.ch verfügbar.

---

19. WELCHE REALISTISCHEN KOSTEN UND RESSOURCEN SIND FÜR EINE CONTAINER-LÖSUNG ZU ERWARTEN?

Die Kosten variieren je nach Grösse und Komplexität des Projekts, dem erforderlichen Schulungsaufwand und ob die Container-Plattform intern oder von einem externen Anbieter verwaltet wird. Eine gründliche Vorabanalyse – idealerweise mit Unterstützung eines erfahrenen externen Partners – hilft dabei, den zu erwartenden Aufwand und die laufenden Betriebskosten realistisch einzuschätzen. Wir bieten Architekturberatung als Teil des Kunden-Onboardings an.

---

20. WIE ÜBERZEUGE ICH MEIN TEAM VON CONTAINERN?

Die Migration zu einem Container-Setup ist mit einem initialen Aufwand und einer Lernkurve verbunden. Am besten baut man Akzeptanz im Team auf, indem man die Vorteile klar aufzeigt: Zeitersparnis durch automatisierte Prozesse, agile Arbeitsmethoden, vereinfachte Skalierung und verbesserte Deployment-Sicherheit. Ein praxisnahes Training – wie die Nine Kubernetes Academy – ist der effektivste Weg, ein Team schnell von der Theorie in die Praxis zu bringen.

---

21. WO FINDE ICH UNTERSTÜTZUNG FÜR CONTAINER- UND KUBERNETES-FRAGEN?

Die offizielle Kubernetes-Dokumentation (kubernetes.io) ist die massgeblichste Quelle für technische Fragen. Die CNCF Landscape (landscape.cncf.io) bietet einen Überblick über das gesamte Cloud-native-Ökosystem. Für Schweizer-spezifische Anforderungen, Architekturberatung oder praktische Unterstützung steht unser Engineering-Team direkt zur Verfügung, nicht über ein anonymes Ticket-System.

Mit einem unserer Engineers über Container-Technologie sprechen

Wir helfen dir gerne weiter, ob du gerade erst anfängst oder einen bestehenden Workload migrieren möchtest.

Kontakt aufnehmen

Nine Internet Solutions AG · Badenerstrasse 47 · 8004 Zürich · Schweiz
info@nine.ch · +41 44 637 40 00 · nine.ch