

Container-Technologie

Eine umfassende Lösung

Diese Übersicht dient dazu, die wichtigsten Fragen rund um die Container-Technologie zu beantworten und einen grundlegenden Überblick über das Thema zu bieten.

Inhaltsverzeichnis.

Die Grundlagen	
Was ist die Container-Technologie oder Container-Virtualisierung?	3
Welche Vorteile bietet eine Container-Lösung?	3
Was sind die Unterschiede zu ‚herkömmlichen‘ Lösungen und Virtuellen Maschinen (VM)?	4
Welche Risiken birgt eine Container-Lösung?	4
Die Implementierung	
Wie verbessert die Container-Technologie die Update- und Implementierungs-Prozesse konkret?	5
Was sind Container-Orchestrierungs-Plattformen?	6
Was ist Kubernetes?	6
Was ist DevOps?	6
Gehören DevOps und Container immer zusammen?	6
Gibt es Best Practices für das Deployment von Container-Umgebungen?	7
Wie gross ist der Lernaufwand für die Arbeit mit Containern?	7
Wie gelingt der Start in ein Container-Projekt?	7
Container-Technologie: managed oder unmanaged?	8
Wo gibt es Tutorials / Antworten auf konkrete anwendungstechnische Fragen im Zusammenhang mit Containern, Kubernetes oder DevOps?	8
Mit welchen Netzwerk-Systemen/-Architekturen sind Container kompatibel? Wo sind die Grenzen?	8
Wie sicher sind Container-Lösungen? Im Vergleich zu virtuellen Maschinen?	8
Wie gewinnt man internes Buy-In für die Einführung von Containern?	9
Welche Kosten und Ressourcen sind für die Implementierung und den Betrieb einer Container-Lösung realistisch?	9
Wie holt man das eigene Team mit ins Boot?	9
Gibt es Alternativen zu Docker?	9

Die Grundlagen der Container-Technologie

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

Docker beschreibt einen Container als eine *standardisierte Software-Einheit*. Die Container-Virtualisierung erlaubt es, eine Anwendung unabhängig von einem Host-Betriebssystem zu betreiben. Im Container verpackt ist also eine komplette Anwendung inklusive Code, Abhängigkeiten und Konfigurationen in einem klar festgelegten Format.

2. WELCHE VORTEILE BIETET EINE CONTAINER-LÖSUNG?

Die Standardisierung eines Containers und seine Unabhängigkeit vom Host-Betriebssystem bietet für die Entwicklung und den Betrieb von Software bestehende Vorteile:

Vorteil #1 Mit Containern lassen sich Applikationen besser skalieren.

Die Container-Virtualisierung basiert auf autonomen, selbständig funktionierenden Einheiten. Damit lassen sich Anwendungen nach Bedarf flexibel und schnell hochfahren oder auch reduzieren.

Vorteil #2 Mit Container laufen Anwendungen schneller.

Container starten schneller, da kein eigenes Gast-Betriebssystem notwendig ist, welches zuerst gestartet und wieder heruntergefahren werden muss.

Vorteil #3 Container setzen vorhandene Ressourcen effizient ein.

Weil Container kein eigenes Betriebssystem benötigen, werden physische Ressourcen eingespart, welche wiederum für andere Anwendungen bereitstehen.

Vorteil #4 Mit Containern lässt sich Software agiler entwickeln.

Einzelne Container lassen sich in kürzester Zeit entwickeln und sind einfach reproduzierbar. Das hat unter anderem zum Vorteil, dass ein in der Entwicklungsumgebung gestarteter Container genau gleich ist wie dieser in der Produktionsumgebung.

Vorteil #5 Container sind portabel.

Da Container Anwendungen weitestgehend unabhängig vom Host-Betriebssystem betrieben werden, lassen sie sich ohne grossen Aufwand migrieren.

Sie wünschen sich eine vertiefte Analyse?

Lesen Sie mehr über die Vor- & Nachteile der Container Technologie in unserem [nine Blog](#) oder laden Sie sich unser Whitepaper "[Container Virtualisierung und ihre Verwendung](#)" herunter.

3. WAS SIND DIE UNTERSCHIEDE ZU "HERKÖMMLICHEN" LÖSUNGEN UND VIRTUELLEN MASCHINEN (VM)?

Container und virtuelle Maschinen (VM) beschreiben zwei unterschiedliche Ansätze, wie man Anwendungen in einer virtuellen Umgebung hosten und betreiben kann. Bei VMs sorgt ein sogenannter Hypervisor für die Verteilung der physischen Ressourcen des Servers in virtuelle Maschinen unterhalb des Betriebssystems. Virtuelle Maschinen benötigen zudem ein eigenes Betriebssystem pro Anwendung. Die Virtualisierung bei Containern passiert im Gegensatz dazu direkt auf der Ebene des Betriebssystems. Dies führt dazu, dass Container die vorhandenen Ressourcen wesentlich effizienter nutzen als virtuelle Maschinen und zudem schneller booten.

[Lesen Sie in unserem nine Blog mehr zu den unterschiedlichen Lösungen in der Container Technologie.](#)

4. WELCHE RISIKEN BIRGT EINE CONTAINER-LÖSUNG?

Wer sich über die Container-Technologie vertieft informiert, der begegnet unter Umständen auch folgenden Vorbehalten:

«Container sind zu wenig sicher für unsere Anforderungen»

Container sind nicht weniger sicher als andere Umgebungen. Wie bei anderen Setups hängt die Sicherheit davon primär ab, wie die Umgebung aufgesetzt und gewartet wird.

«Die Container-Verwaltung ist viel zu komplex für uns»

Wie bei jeder Veränderung ist auch die Umstellung auf Container mit einem gewissen initialen Aufwand verbunden. Applikationen müssen vorbereitet, Mitarbeitende trainiert und gegebenenfalls Deployment- und Arbeitsprozesse automatisiert werden. Doch diese Investition lohnt sich langfristig und wird durch die oben aufgeführten Vorteile der Containerisierung schnell wettgemacht. Mit Unterstützung von Experten lässt sich dieser Aufwand weiter reduzieren.

«Wir haben eine Deployment-Strategie. Wir brauchen keine Container»

Container ersetzen keine Deployment-Strategie sondern ergänzen und optimieren diese. Die Implementierung der Container-Technologie bedingt nicht, dass Sie Ihr bisheriges Setup komplett umkrempeln. Es gibt diverse Möglichkeiten, Container in eine bestehende IT-Strategie zu integrieren – zum Beispiel in Kombination oder ergänzend zu virtuellen Maschinen.

«Container sind nur ein Trend. Wir setzen nur auf Technologien von Dauer»

Die Anfänge des Container-Prinzips liegen im Jahr 1979. Seit der Einführung von Docker im Jahr 2013 hat sich die Container-Technologie konstant weiterentwickelt und ist mittlerweile weit verbreitet und etabliert. So verwalten grosse Tech-Unternehmen wie Amazon, Google und PayPal ihre Anwendung mittels Container und optimieren damit ihre Deployment-Prozesse.

So implementieren Sie Container-Technologie

5. WIE VERBESSERT DIE CONTAINER-TECHNOLOGIE DIE UPDATE- UND IMPLEMENTIERUNGS-PROZESSE KONKRET?

Bei Update- und Implementierungs-Prozessen sind die Geschwindigkeit und die Fehlerfreiheit des Codes wichtige Messwerte. Und gerade hier fallen die Vorteile von Containern am stärksten ins Gewicht.



Beispiel: Geschwindigkeit als Messwert

- Schnelleres Booting
- kürzere Feedback-Zyklen
- Einfacheres Scaling (Es ist nur notwendig, die Konfiguration zu setzen und die Plattform übernimmt diese automatisch)
- Rollende Deployments ohne Unterbrüche (Neue Versionen sind sofort verfügbar)

6. WAS SIND CONTAINER-ORCHESTRIERUNGS-PLATTFORMEN / CLUSTER MANAGER?

Eine Container-Orchestrierungs-Plattform dient dazu, die verschiedenen Container zu verwalten. Sie ist ein dynamisches System, welches Container gruppiert, platziert, deren Verbindungen managt und die Kommunikation sicherstellt. Eine solche Plattform (oder Cluster) kann "on-premise" oder auch auf gängigen Cloud-Plattformen (z.B Google Cloud) betrieben werden. Die Orchestrierung der Container wird zum Beispiel mit OpenShift oder Kubernetes gemanagt.

7. WAS IST KUBERNETES?

Kubernetes dient der Orchestrierung von mehreren Containern. Die Open-Source-Plattform automatisiert also das Einrichten, Betreiben und auch das Skalieren von containerisierten Anwendungen.

8. WAS IST DEVOPS?

DevOps baut als Organisations- und Führungskultur auf drei Prinzipien auf - besser bekannt als «The Three Ways»: Dem Flow-Prinzip, dem Feedback-Prinzip und dem Prinzip des kontinuierlichen Lernens und Experimentierens. Diese drei Wege sind bei der Umsetzung zentral, um das übergreifende Ziel - besser und schneller werden - zu erreichen. Dabei dienen die drei Wege als Rahmen für Prozesse, Verfahren und Praktiken, bei denen ganz klar die Optimierung der Zusammenarbeit im Vordergrund steht.

Lesen Sie hier mehr über [«The Three Ways»](#) - die Prinzipien von DevOps und warum auch andere Teams eines Unternehmens davon profitieren können.

9. GEHÖREN DEVOPS UND CONTAINER IMMER ZUSAMMEN?

DevOps als Philosophie und die Container als Technologie spielen optimal zusammen. Sie sind jedoch nicht voneinander abhängig, was bedeutet, dass DevOps ohne Container ebenso möglich ist, wie die Container-Virtualisierung ohne DevOps. Die Docker Deployment-Strategie rät generell zu möglichst wenigen manuellen Eingriffen (Automatisierung), um die Bewirtschaftung von Container möglichst reibungslos zu gestalten und alle Vorteile zu nutzen. Der DevOps-Gedanke kann durchaus auch in der Arbeit mit klassischen Servern gelebt werden, zum Beispiel wenn es darum geht, die organisatorischen Prozesse zu vereinfachen.

Setzen Sie auf Docker und Kubernetes, wenn Sie mit Containern starten wollen

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

Der erste Einstiegspunkt zu Container-Orchestrierung ist meist Docker-compose. Dieses Tool ist jedoch primär für die lokale Entwicklung geeignet und eignet sich nicht für die Verwendung in Container-Orchestrierungs-Plattformen.

Für Kubernetes gibt es Tools wie Kompose und compose-on-kubernetes, welche die Docker-Compose-Files in API-Objekte übersetzen. Ein vertieftes Verständnis dieser Objekte ist jedoch zwingende Voraussetzung, um bei Problemen selbständig Trouble-Shooting zu betreiben und Installationen zu debuggen.

11. WIE GROSS IST DER LERNAUFWAND FÜR DIE ARBEIT MIT CONTAINERN?

Hier spielt der Grad der gewünschten Autonomie eine Rolle. Wenn der Betrieb der Containerplattform komplett ausgelagert wird, reichen grundlegende Kenntnisse in Bezug auf die Plattform-API aus. Wird der Betrieb in-house sichergestellt, erfordert der Umstieg auf eine Container-basierte Architektur eine umfangreiche Umschulung von Entwicklern und System Engineers. Ohne ein fundiertes Wissen über die Anwendung von Docker und der gewählten Orchestrierungsplattform geht es nicht.

12. WIE GELINGT DER START IN EIN CONTAINER-PROJEKT?

Eine optimale Grundlage bietet zum Beispiel ein Basis-Workshop zu Docker und Kubernetes. Denn diese Kenntnisse stellen eine Grundvoraussetzung für das Verständnis zu Container dar.

13. CONTAINER-TECHNOLOGIE: MANAGED ODER UNMANAGED?

Dies kommt ganz auf die digitale Strategie Ihres Unternehmens an. Je nach Setup macht es mehr oder weniger Sinn, die Plattform intern zu betreiben.

14. WO GIBT ES TUTORIALS UND ANTWORTEN AUF KONKRETE ANWENDUNGSTECHNISCHE FRAGEN IM ZUSAMMENHANG MIT CONTAINERN, KUBERNETES ODER DEVOPS?

Die Blogs von Kubernetes und Docker sind gute Anlaufstellen für konkrete Fragestellungen. Auf diesen und auch vielen weiteren Experten-Blogs befinden sich zudem diverse Online-Tutorials, die schnell Antworten bei konkreten Problemstellungen liefern. Auch bei nine sind hilfreiche Informationen im Support-Bereich zu finden.

15. MIT WELCHEN NETZWERK-SYSTEMEN/ -ARCHITEKTUREN SIND CONTAINER KOMPATIBEL? WO SIND DIE GRENZEN?

Container können eine sinnvolle Ergänzung oder Alternative zu VMs sein. Grundsätzlich können Container auch innerhalb eines Netzwerk mit VMs verbunden werden, wenn es die Ausgangslage erfordert. Wichtig: Container laufen sowohl auf Windows als auch Linux, wobei die Windows-Option noch deutliche Nachteile im Gegensatz zu Linux aufweist.

Lesen Sie im [Interview mit unserem Platform-Teamleiter Tom Whiston](#), wie die Container die Sicherheit in der Cloud verbessern können.

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

Container sind genau so sicher wie traditionelle Serverlösungen und VMs. Zentral ist, wie diese Umgebungen aufgesetzt werden. Die Fragestellung bezüglich Betrieb und Sicherheitsvorkehrungen unterscheiden sich lediglich im Sicherheitsmodell. In jeder Architekturebene kann man sinnvolle Varianten wählen, um Sicherheit zu gewährleisten - unabhängig davon, ob man Container vor Ort oder in der Cloud betreibt. Letztlich hängt die Sicherheit stark davon ab, wie sicher die Anwendung selbst ist.

Lesen Sie in unseren [Case Studies](#), wie unsere Kunden die Zusammenarbeit mit uns als externen Experten im Bereich Cloud Computing und Container erlebt haben.

17. WIE GEWINNT MAN INTERNES BUY-IN FÜR DIE EINFÜHRUNG VON CONTAINERN?

Wie bei anderen Neuerungen und Investitionen überzeugt man auch bei der Diskussion um die Einführung von Containern letztlich mit dem konkreten Aufzeigen von Kosten-Einsparungen und einer verbesserten Time-to-Market. Erfahrungsgemäss helfen dabei praktische Case Studies und Fallbeispiele enorm.

18. WELCHE KOSTEN UND RESSOURCEN SIND FÜR DIE IMPLEMENTIERUNG UND DEN BETRIEB EINER CONTAINER-LÖSUNG REALISTISCH?

Die Kosten variieren nach der Grösse und Komplexität des Projekts, des Trainingsaufwands, ob die Containerplattform auf der eigenen Infrastruktur oder ausserhalb betrieben wird. Daher ist es wichtig, vorab eine detaillierte Situationsanalyse – allenfalls unter Beizug eines externen Experten - durchzuführen um den anstehenden Aufwand einzuschätzen.

19. WIE HOLT MAN DAS EIGENE TEAM MIT INS BOOT?

Der Wechsel auf ein Container-Setup kann mit einem hohen Erst- und Lernaufwand verbunden sein. Die Akzeptanz im Team kann man am besten damit erreichen, indem man die Vorteile der Container-Lösung klar aufzeigt: Zeiterparnis durch automatisierte Prozesse, agile Arbeitsweisen, Skalierbarkeit etc.

20. GIBT ES ALTERNATIVEN ZU DOCKER?

Dank des Open-Source-Charakters der Container-Technologie gibt es immer wieder neue Tools, die Anwendungen von Docker ersetzen. Beispiele sind PodMan und Buildah, ersteres ersetzt die Docker Command line und ist für die Betreuung von standalone-Container geeignet. Buildah, ein Shell-basiertes Linux-Tool, ermöglicht das Erstellen von Container-Images. Daneben gibt es diverse Run-time-Lösungen wie CRI-O, die anstelle von Docker als Runtime auf Kubernetes verwendet wird.

Buchen Sie jetzt ein Gespräch mit einem nine Experten im Bereich der Container-Technologie...

...und erörtern Sie mit uns, wie Sie von der Container-Technologie profitieren können.

TERMIN BUCHEN

Das könnte Sie auch noch interessieren:

WHITEPAPER «CONTAINER-VIRTUALISIERUNG UND IHRE VERWENDUNG»



Was genau es mit der Container-Technologie auf sich hat und wie Sie davon profitieren können

- Alles was Sie über Container wissen sollten
- Deployment-Prozesse - klassisch vs. containerisiert
- Prozessoptimierung dank Containern
- So kann eine CI/CD-Pipeline aussehen
- Einfluss von Docker und OpenShift auf Development-Prozesse

JETZT HERUNTERLADEN