

## Eine hochverfügbare und resiliente Plattform, um Container mit weniger Verwaltungsaufwand zu betreiben

Während Scale Computing immer häufiger in dezentralisierten Unternehmen eingeführt wird, sehen wir zugleich eine verstärkte Einführung von Containern und den Einsatz API-getriebener Automatisierung wie Infrastruktur als Code. Große Organisationen haben oft Workloads, die innerhalb von Containern laufen müssen. Die Containerisierung wird von vielen Faktoren begünstigt: moderne Anwendungen werden oft als „Container first“ entwickelt, Microservices werden zunehmend verteilt, und hybride Cloud-Verteilungen verlangen portable Workloads.

### Nahtlose programmatische Verteilung

Scale Computing HyperCore (SC//HyperCore) ermöglicht eine nahtlose programmatische Verteilung von Containern. Um Container auf SC//HyperCore laufen zu lassen, verteilen Nutzer einfach ein containeroptimiertes Betriebssystem mit einer Container-Laufzeit der Wahl (häufig Docker oder in einer Kubernetes-Umgebung, in Containern oder CRI-O). Dies lässt sich manuell bewerkstelligen, allerdings erfordern Anwendungsfälle für Container häufig einen hohen Grad der Automatisierung. Aus diesem Grund revolutionieren unsere REST-APIs und der Support von Cloud-init, wie Nutzer containerisierte Workloads laufen lassen können, indem sie die Installation des Betriebssystems, die Container-Laufzeit und den Workload automatisieren.

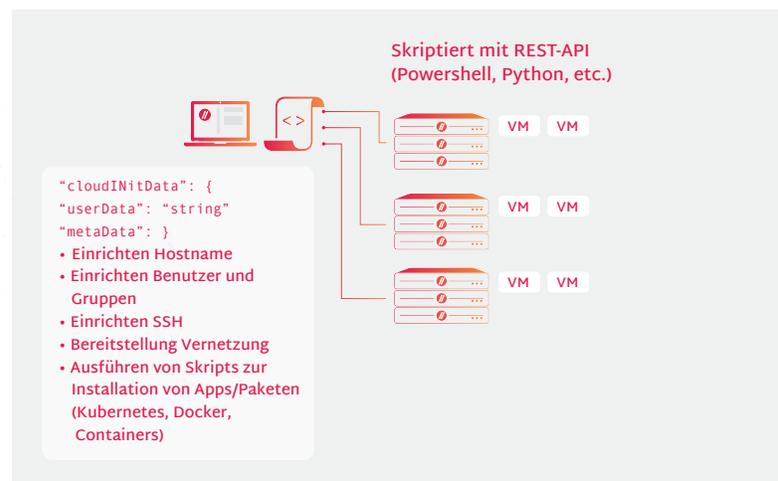
### Cloud-init Anpassung mit REST-APIs

SC//HyperCore beinhaltet die Cloud-init Anpassung durch REST-APIs, um die Infrastruktur als Code zu ermöglichen, damit Entwickler und Administratoren ansonsten hochgradig manuelle Prozesse automatisieren können.

Cloud-init in SC//HyperCore ermöglicht die einfache und massenhafte Bereitstellung kundenspezifischer VMs.

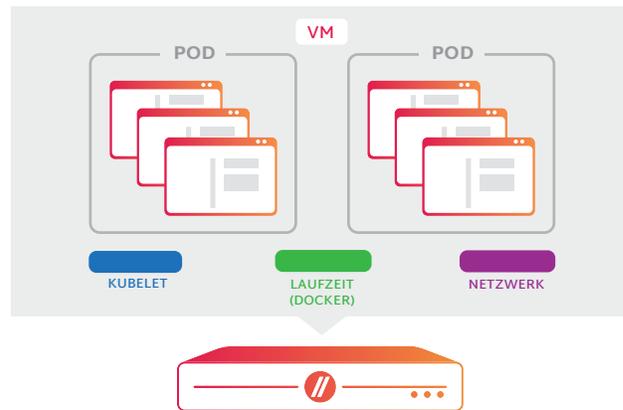
#### Vorteile:

- Zeitersparnis durch Automatisierung manueller Schritte bei der Einrichtung
- Reduzierung menschlicher Fehler bei manueller Einrichtung
- Gewährleistung einheitlicher Verteilung an verschiedenen Standorten
- Einheitliche Änderungssteuerung und zuverlässigere Updates durch Standardisierung update through standardization



Scale Computing ist sich bewusst, dass SC//HyperCore sowohl bei der Unterstützung von VMs als auch von Containern fehlerfrei arbeiten muss. Es muss einfach zu verwalten sein und Container unabhängig von der Implementierung programmatisch verteilen. Die jüngsten Investitionen in REST-APIs beschleunigen und erleichtern die Verteilung von Masse-Containern auf SC//HyperCore enorm. Somit können Organisationen Verwaltungsaufwand reduzieren, indem sie Skripte schreiben, um die täglichen Verwaltungsaufgaben bedarfsgerecht zu erledigen, vor allem bei der Verteilung von Containern. Vielleicht der wichtigste REST-API-Endpunkt für die Verteilung von Containern auf SC//HyperCore ist CloudInitData. Cloud-init ist eine leistungsfähige Open-Source-Technologie, die es Nutzern erlaubt, VMs beim ersten Booten automatisch zu konfigurieren. Insbesondere ist Cloud-init ein Paket, das für die meisten Linux-Distributionen verfügbar ist und Nutzern erlaubt, ein Cloud-Image (im Grunde ein ganz normales, mit Cloud-init versehenes Linux-Image) aufzunehmen und die VM während des ersten Systemstarts über ein Skript mit Meta- und Nutzerdaten zu versorgen. Die verfügbaren Konfigurationen sind sehr leistungsstark. Nutzer können SSH-Schlüssel einrichten, Nutzer anlegen, Dateien schreiben und sogar ganze Befehle automatisch ausführen.

Run Docker - oder jede andere Container-Laufzeit - steuert unmittelbar nach dem Booten der VM die Erstellung von Containern innerhalb einer VM, ohne dass der Nutzer manuell eingreifen muss. Container innerhalb einer VM auszuführen, bietet viele Vorteile, insbesondere wenn man Container ohne eine Containercluster-Verwaltung wie Kubernetes verteilt. SC//HyperCore bewältigt die Herausforderungen beim Einrichten einer hohen Verfügbarkeit, die mit jeder Verteilung von Containern einhergehen.



## Kubernetes Distribution

Heute lässt sich fast jede Kubernetes-Distribution innerhalb von VMs, die auf SC//HyperCore laufen, problemlos verteilen. Unsere REST-APIs und der Cloud-init-Support verbessern das Tempo und die Effizienz bei der Bereitstellung neuer K8S-Cluster und Anwendungen durch Infrastruktur als Code.

Alle Kubernetes-Verteilungen (sowie alle Container-Verteilungen) nutzen unsere redundante Speicher- und Rechnerarchitektur, um die Verfügbarkeit der Anwendung sicherzustellen. Aus diesem Grund können sogar Kubernetes-Cluster aus einzelnen Knoten auf SC//HyperCore verteilt werden und volle Resilienz für zustandsbezogene Anwendungen und Daten wie auch für die Steuerebene von Kubernetes / den API-Server selbst bieten, was die Komplexität und die notwendigen Ressourcen für die Verteilung von Kubernetes an Edge-Standorten deutlich reduziert. Wenn ein Kubernetes-Cluster aus vielen Knoten zustandsbezogene Anwendungsdaten über den K8S-Cluster teilen muss, können Kunden Container Attached Storage nutzen, indem sie Kubernetes-Speichererweiterungen wie OpenEBS einsetzen. Außerdem kann sich Kubernetes mit einem externen Speichergerät wie NAS/SAN verbinden oder ein virtuelles NAS innerhalb einer resilienten VM ausführen, um NFS oder eine Objektspeicherung zu bedienen.

SC//HyperCore ist für diesen Anwendungsfall bereits unglaublich leistungsfähig; dennoch stellt Scale Computing weitere Funktionen bereit, um die Container noch einfacher verteilen und verwalten lassen. So kann die Einbindung in verschiedene Drittanbieter-Tools wie IBM Edge Application Manager, Avassa und Portainer den Nutzern ebenfalls helfen, Container zentral zu verwalten, zu verteilen und zu aktualisieren. Wir werden auch weiterhin in die Zukunft der Verteilung von Anwendungen investieren.