MAGAZIN FÜR PROFESSIONELLE
INFORMATIONSTECHNIK

Mit IBMs z15 und z/OS 2.4: zCX Container sicherer betreiben

18.12.2019 09:00 Uhr Christoph Puppe

Eine zentrale Neuerung bei IBMs aktueller Betriebssystem-Generation z/OS 2.4 sind die Container Extensions. Wir zeigen potenzielle Einsatzmöglichkeiten.

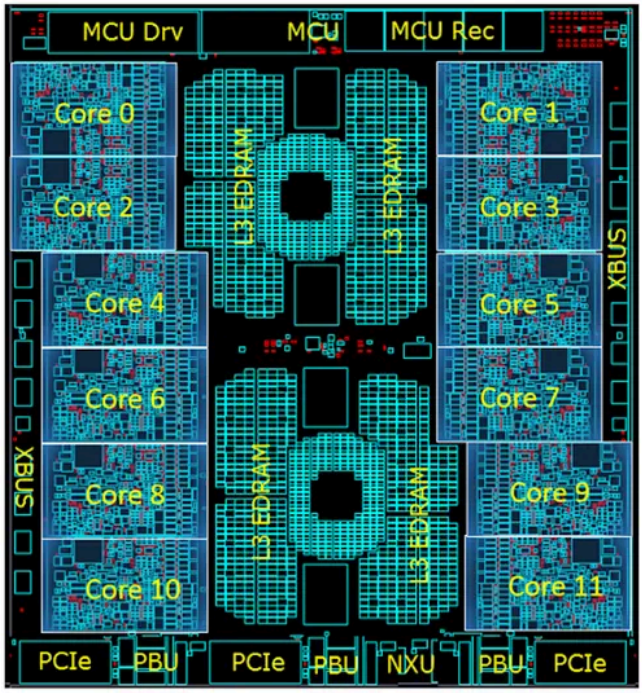
Seit September 2019 liefert IBM seine neue Mainframe-Generation z15 mit dem auf Version 2.4 aktualisierten z/OS aus. Neu in dieser Release sind unter anderem die Container Extensions (zCX). Darüber lassen sich Docker-Container, die unter Linux on Z bereits in verschiedenen Umgebungen laufen, nun auch direkt neben z/OS-Middleware wie CICS, IMS oder der Datenbank DB/2 betreiben.

Damit eignet sich zCX besonders für mit dieser Middleware verbundene Anwendungen. Zielgruppe sind vor allem sehr große Umgebungen mit hohen Anforderungen an Skalierung, Verfügbarkeit und Sicherheit. IBM gibt an, dass die z15 bis zu 2,4 Millionen Container gleichzeitig verkraftet, und hat die Separierung der Umgebungen nach EAL5+ zertifiziert. Auch für das Vorgängermodell z14 will IBM z/OS 2.4 und damit zCX anbieten.

Kunden, die heute ihre Reise auf dem Mainframe beginnen, kommen oft aus der Linux-Welt. Dieses Wissen können sie mit Docker und Kubernetes auf Linux on Z einsetzen. Denjenigen, die bereits Erfahrung mit z/OS haben, aber nicht tief ins Linux-Ökosystem einsteigen wollen, bietet zCX die

Möglichkeit, diese Erfahrungen direkt zu nutzen und Softwarekomponenten aus dem Linux-Umfeld mit den eingeführten Betriebsprozessen zu installieren und zu verwalten. Aktuell interessieren sich hauptsächlich Firmen und Behörden für zCX, die bereits Mainframes im Einsatz haben.

In der z15 ist der Beschleunigerchip für die Kompression der Daten als NXU in den Prozessor-Die gewandert. Dort erhält er die Daten direkt aus dem L3-Cache und steigert seinen Durchsatz auf 275 GByte/s. Insgesamt verfügt eine vollausgestattete z15 jetzt über 40 TByte statt 32 TByte Speicher und rechnet mit 190 statt 170 Cores. IBM liefert die neuen z15-Konfigurationen erstmals in 19-Zoll-Racks statt im bisherigen Format der z14, das aus dem Rack hervorstand, oder dem alten proprietären Format. Eine z15 ist für ein bis vier Racks ausgelegt, die von der Größe und den Anschlüssen her unproblematisch sind und ohne besondere Anforderungen in die üblichen RZ-Ausbauten passen.



Eine in 14-nm-Technologie gefertigte z15-CPU bietet zwölf Rechenkerne.
(Bild: IBM)

IBM liefert seine seit September 2019 verfügbaren z15-Systeme erstmals als Standard-19-Zoll-Racks aus.
(Bild: IBM)

Auf der Z-Serie läuft der Standard-Kernel der eingesetzten Linux-Distribution. Das heißt, eine Anwendung, die unter Linux läuft, wird dies auch auf Linux on Z tun. Letzteres steht dabei entweder für SUSEs SLES, Red Hats RHEL oder Canonicals Ubuntu. Allerdings ist der oben beschriebene z15-Prozessor kein x86 und Anwendungen müssen für die 64Bit-s390x-Architektur kompiliert sein.

Z-Linux und LPAR

Container-Images für die Architektur finden sich in Container-Repositories wie dem Hub von Docker mit dem Tag s390x oder "IBM Z". Programme, die in Java, Python, JavaScript et cetera vorliegen, können auf den existierenden Docker-Base-Images aufbauen und Entwickler damit ohne Änderungen des Dockerfiles ihre Images bauen. Für alle, die Performance mögen und in Go, C, C++ et cetera schreiben, gibt es native und Cross-Compiler. Anpassungen des Codes sind nur selten notwendig, wenn auch wegen der unterschiedlichen Endianness nicht ausgeschlossen. Zumeist sind es die Build-Parameter, die eine Änderung benötigen.

Auf IBMs Mainframes ist es seit über 50 Jahren üblich, Anwendungen in Partitionen zu starten. Im DPM (Dynamic Partition Manager) konfiguriert der Administrator – System Programmer genannt – den Hypervisor PR/SM, eine Weiterentwicklung von CP-40. Der ist seit 1972 der Typ-1-Hypervisor, der die Hardware eines Mainframes aufteilt. Im allgemeinen Sprachgebrauch hat sich LPAR (Logical Partition) als Name dafür durchgesetzt – sie hat außer dem Begriff Partition nichts mit den Dateisystempartitionen klassischer Intel-Architekturen zu tun.

LPARs lassen sich mit HyperSwap zwischen Speichersystemen an unterschiedlichen Standorten sichern und dort neu starten; z/VM kann Gäste zwischen Hosts verschieben. Beides hilft, den Ausfall oder eine vollständige Abschaltung einer z15 ohne Unterbrechung der Prozesse zu überstehen oder sehr schnell wieder zu starten. LPARs und zVM sind die Separierungstechniken, die bei fast allen Geldinstituten, Zentralbanken und Kreditkartenfirmen die Sicherheit der Anwendungen vor Ausfällen und Fremdeinfluss garantieren.

[1]

Eine z15 beherrscht bis zu 80 LPARs. In denen läuft entweder ein zOS einzeln oder das System ist durch z/VM in weitere virtuelle Instanzen aufgeteilt. Sofern das Betriebssystem der LPAR "Linux on Z" ist, lassen sich dort per KVM weitere Linux-Kernels starten. Wie man es auch von anderen

Virtualisierern kennt, sieht zVM ein virtuelles Netzwerkinterface, Festplatten und Speicher vor. Hier stellt der native Speichertreiber, den die LPAR nutzt, bei zVM die Festplatten der Gäste als VSAM-Datenset aus dem extern per FICON oder Fibre Channel angeschlossenen Blockspeicher bereit. IBM bietet mit dem DS8900F, das bis zu 4,5 PByte und Zugriffszeiten von minimal 18 s liefert, passende Geräte an. Aber auch Storage anderer Hersteller wie DELL PowerMax ist über FICON anschließbar.

Der Speicher ist auf Wunsch verschlüsselt und komprimiert. Das erfolgt durch dedizierte Hardware in der CPU – dank des Zugriffs auf den L3-Cache mit nur geringen Auswirkungen auf den Durchsatz. Immutable Snapshots, also eine Versionierung des kompletten Speichers gegen ungewollte Veränderungen, und Replikationen innerhalb einer Zone oder zu anderen Rechenzentren sind möglich.

Intern sind die Netzwerke über einen zwischen LPAR und zVM geteilten Speicher realisiert. Das Shared Memory Communications Direct Protocol (SMC-D) regelt dabei den Zugriff der einzelnen Partitionen auf die Speicherseiten, die es den Anwendungen als TCP/IP-Datenverkehr präsentiert. Die Firmware simuliert den Netzwerkverbindungsaufbau als Zuteilung geteilter Adressen im physischen Speicher und beschleunigt so Durchsatz, Datenmenge und Latenz. Mit Remote Direct Memory Access (RDMA) ist dies als SMC-R auch zwischen mehreren Hosts in einem Netzwerk möglich. Eine Besonderheit ist hier die schon erwähnte Kompression und Verschlüsselung der Daten vor der Übertragung bereits auf dem Chip.

[2]

GLOSSAR

- **CICS**: Customer Information Control System, Transaktionsserver für z/OS
- **CP-40**: Control Program, mit der IBM 360 Model 40 eingeführter Hypervisor zum logischen Partitionieren der Hardware
- **DPM**: Dynamic Partition Manager
- **FICON**: Fibre Connection, das IBM-eigene Level-4-Protokoll dient wie Fibre Channel zur Storage-Anbindung.
- **IMS**: Information Management System, ein nachrichtenbasierter Transaktionsmanager und hierarchischer Datenbankmanager für z/OS
- **LPAR**: logische Partitionen auf IBM-Hosts
- **NXU**: Nest Acceleration Unit, Hardwarebeschleuniger für Kompressionsalgorithmen, On-Chip-Nachfolger der Enterprise Data Compression zEDC
- **PR/SM**: Processor Resource / System Manager, IBMs Typ-1-Hypervisor für Mainframes
- **SSC**: Secure Service Container

- **SMC-D**: Shared Memory Communications – Direct Memory Access, ein internes Netzwerk auf IBM Mainframes, das über geteilten RAM die Daten nicht kopiert, sondern den Empfängern als Adresse im RAM bereitstellt.
- **SMC-R**: Shared Memory Communications – Remote Memory Access, SMC-Variante für RDMA
- **Sysplex**: System Processing Complex, eine lose Rechnerkopplung von Großrechnern
- **RDMA**: Remote Direct Memory Access, erlaubt den direkten Zugriff auf Speicher im RAM eines anderen, über ein Netzwerk verbundenen Computers.
- **VSAM**: Virtual Storage Access Method, IBMs Technik zum Dateizugriff auf Blockspeicher
- **zCX**: z/OS Container Extensions
- **z/OS**: natives Betriebssystem der IBM-z-Serie
- **z/OSMF**: z/OS Management Facility, Web-GUI mit REST-API für Systemmanagementfunktionen
- **z/VM**: virtuelle Partitionen innerhalb einer LPAR

Z Container Extensions

Um Container auf einer z15 zu starten, gibt es stark vereinfacht zwei Wege – ein Host kann beide parallel in jeweils mehreren Instanzen betreiben. Der erste ist, ein Linux auf Z oder LinuxONE zu installieren. Dieses kann dann einen Container-Daemon wie dockerd mit oder ohne Orchestrierung durch Swarm oder Kubernetes starten oder über KVM mehrere Linux-Kernels mit eigenen Container-Daemons. Bei dieser Option sind dann das Betriebssystem und die Container wie auf allen anderen Distributionen auch zu betreiben und der Administrator hat vollen Zugriff. Für den Betrieb mit vorhandener Linux-Erfahrung ist diese Variante die empfohlene.

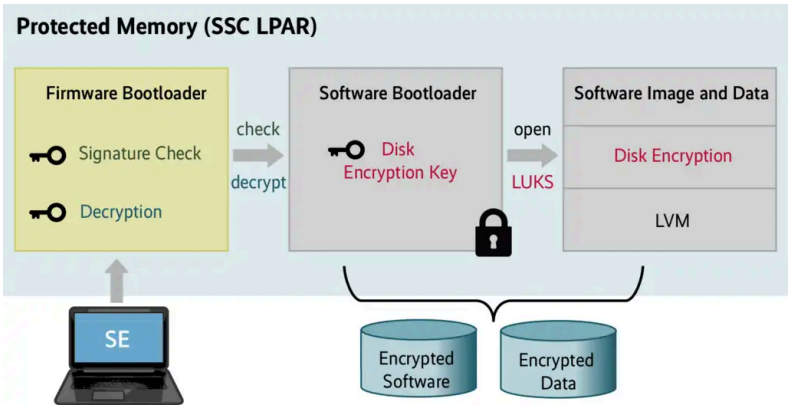
Der zweite Weg sind die seit z/OS 2.4 erhältlichen Z Container Extensions (zCX). zCX laufen als Anwendung in z/OS und können direkt auf dessen Features zurückgreifen. Sie sind eine Containerlaufzeitumgebung, vergleichbar mit einem Linux on Z, auf dem bereits ein Docker-Daemon installiert ist und das auf Ubuntu aufbaut. zCX legen dabei den Fokus ausschließlich darauf, Docker-Container auszuführen, und bieten keine generische Linux-Umgebung. Die Kommandozeile ist dementsprechend auf die zum Betrieb von Containern notwendigen Befehle reduziert. zCX setzt die Installation der Hardwareoption "Container Hosting Foundation" (Feature Code 0104) im Host voraus.

Der Administrator kann in zCX wie auch in den auf Linux on Z laufenden Container-Daemons alle mit s390x oder "IBM Z" getaggten Images vom Docker Hub starten. Aktuell gibt es über 2800 davon. Auch die üblichen Base-Images sind vorhanden, eigene Images zu erstellen ist also kein Problem. Sofern das Base-Image eine s390x als Plattform unterstützt, sind keine Änderungen notwendig, da Docker schon beim Download automatisch die richtige Architektur auswählt.

zCX ist nicht zu verwechseln mit "Secure Service Container" (SSC). SSC ist eine von IBM entwickelte, sogenannte Secure Enclave Technology, um Workloads vor Administrator- und Operator-Zugriffen

schützen zu können. Dies zielt besonders auf das Auslagern von Anwendungen zu Dienstleistern ab. Letztere sollen keine Rechte auf die Anwendungen und die Daten erhalten, sondern nur die Hardware betreiben. Der Administrator des Host-Betreibers kann den SSC nur starten oder stoppen und hat keinen Einblick in die darin laufenden Anwendungen oder Daten. Ein SSC ist dabei eine spezielle LPAR-Variante und somit eine Partition des Hosts mit Linux on Z (mit oder ohne KVM), die wiederum Host für Tausende von virtuellen Gästen sein kann. Jeder dieser Gäste kann dann wiederum beispielsweise ein Linux on Z mit einem Container Daemon und andere Anwendungen enthalten.

SECURE SERVICE CONTAINER



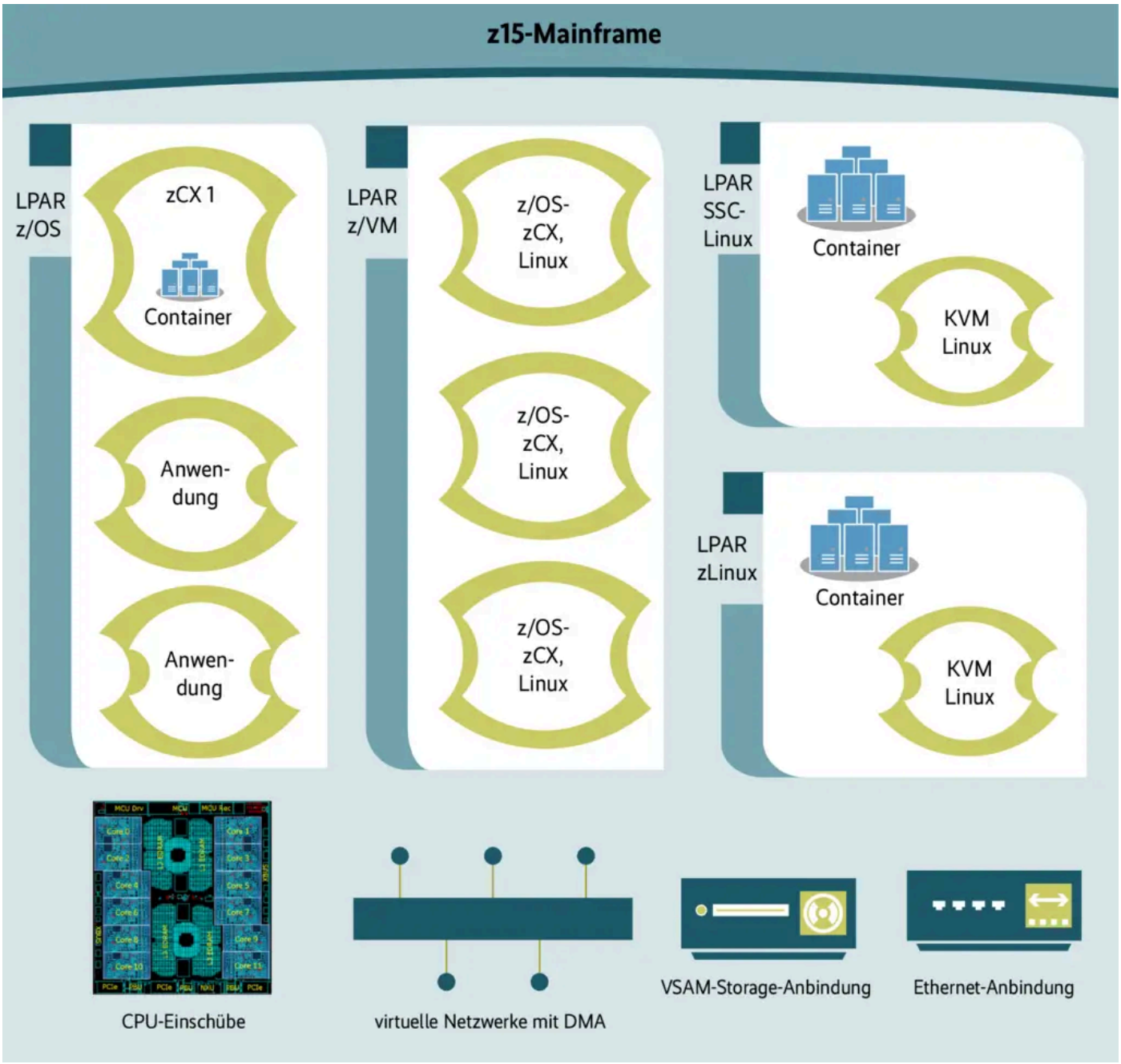
Beim Secure Boot einer SSC-Appliance befinden sich auf dem Storage ausschließlich verschlüsselte Daten. (Bild: IBM)

SSC – Secure Service Container – sind ein IBM-eigenes Format für Container auf z/OS. Wie von Docker-Containern her bekannt, definiert ein SSC alle für den Betrieb der enthaltenen Anwendung notwendigen Einstellungen, Ressourcen und Schnittstellen. Er erlaubt das schnelle Installieren definierter Softwarepakete. Wie ein Container unter Linux trennt SSC die enthaltene Anwendung von anderen auf dem Host laufenden Prozessen. IBM geht mit SSC allerdings einige Schritte weiter: Eine durch SSC installierte Anwendung läuft als Appliance, die für Administratoren auf dem Mainframe nicht mehr erreichbar ist. Umgesetzt hat man diese Separierung über folgende Maßnahmen:

- Kein Zugriff auf die Inhalte des SSC für Administratoren.
- Kein Zugriff per SSH oder andere CLI-Tools in die Appliance.
- Nur Zugriffe von außerhalb des Mainframes sind gestattet.
- Der Zugriff auf das RAM der Appliance ist für alle Anwendungen auf dem Mainframe blockiert.
- Der Datenverkehr zum Storage oder zu anderen Rechnern im Netzwerk wird in der Appliance verschlüsselt.
- Alle Daten-Dumps für das Debugging sind verschlüsselt.
- Die Isolation ist nach dem LinuxONE EAL5+ Protection Profile zertifiziert.
- Images für die Appliances sind verschlüsselt und signiert und werden on the fly mit LUKS entschlüsselt, liegen also immer verschlüsselt auf dem Storage.

Ein SSC ist auf 10 TByte RAM pro Partition limitiert.

Alle diese Optionen haben jeweils Vor- und Nachteile. Ihnen gemeinsam ist, dass die jeweiligen Adressräume entweder durch LPAR, z/VM oder KVM getrennt sind. Die Zuteilung von Ressourcen wie Netzwerkblöcken, Firewalls und Routing erfolgt über die jeweiligen Mechanismen. Somit hängt auch die Isolierung der Instanzen von den gewählten Optionen ab. Den Durchgriff auf die Hardware gewährleistet dabei in allen Fällen die Virtualisierungsschicht von PR/SM. Die Abbildung veranschaulicht in vereinfachter Form die Betriebsoptionen.

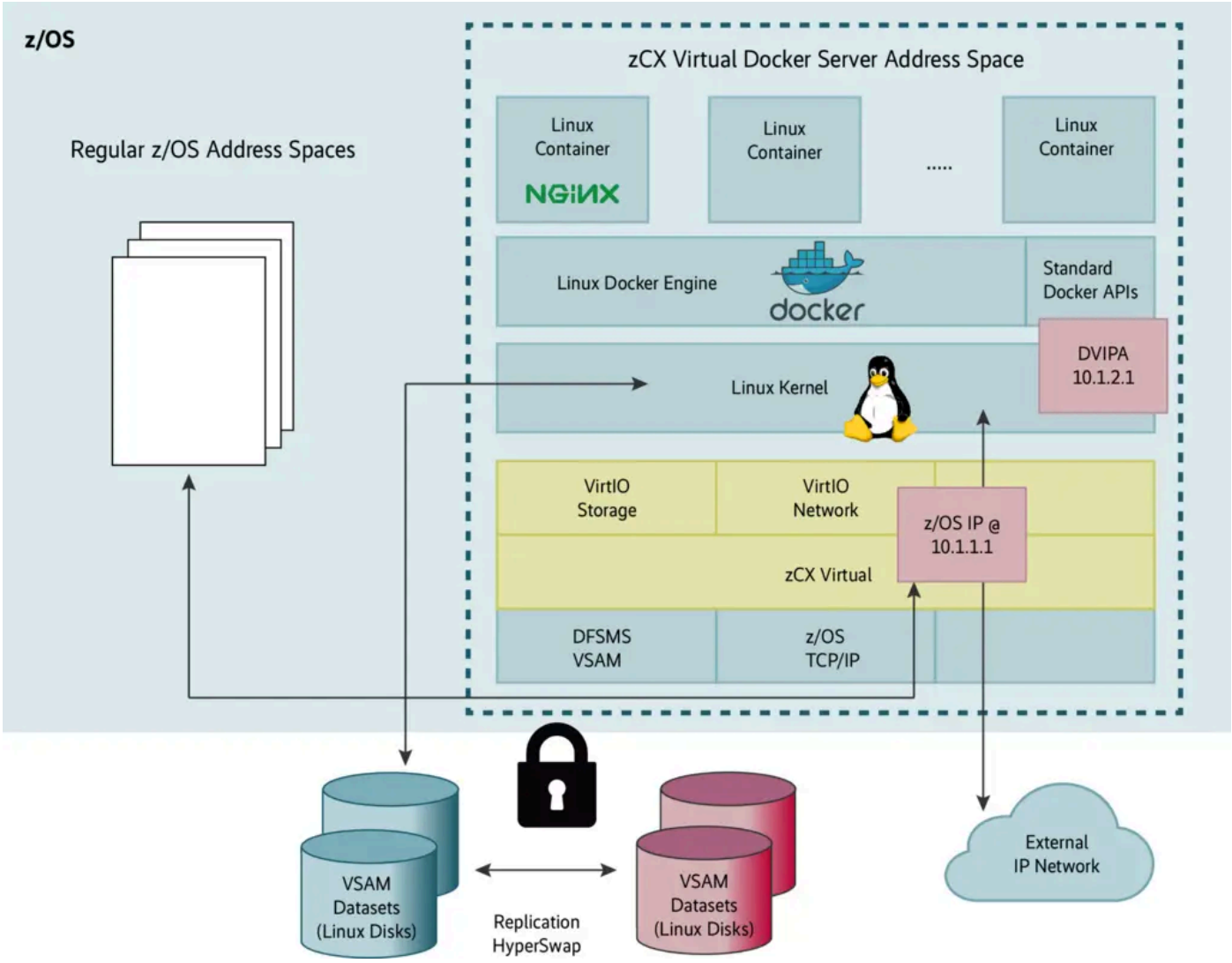


zCX schaffen Flexibilität auf dem Mainframe: Die stark vereinfachte Darstellung zeigt die häufigsten Betriebsoptionen. zCX ist hier mehrfach dargestellt, aber jede Option kann natürlich innerhalb einer LPAR oder in anderen LPARs mehrfach laufen.
(Bild: IBM)

Linux für Container

Die Installation von zCX, dem auf den Containerbetrieb optimierten Linux on Z, mit allen Zuweisungen von Netzwerken, Speicher und CPU erfolgt auf z/OS mit einem z/OSMF-Arbeitsprozess. Von anderen Installationsmethoden rät der Hersteller explizit ab.

Wer sich auf einem zCX anmeldet, erhält nur eine eingeschränkte Shell. Der Kernel und die meisten Funktionen sind hier nicht zugänglich, sondern es stehen nur die Befehle zum Steuern des Docker-Daemons zur Verfügung. Gedacht ist zCX für den Betrieb von Linux-Anwendungen unter z/OS, ohne sich um die Wartung und den Betrieb von Linux on Z kümmern zu müssen. Innerhalb von zCX können entweder einer oder mehrere Container laufen. Steuern lassen sich diese entweder manuell über die Docker-Kommandozeile oder über einen Orchestrator. Aktuell bietet IBM für die zCX nur Swarm für die Orchestrierung an, Kubernetes ist geplant. Mit dem Verkauf von Docker Enterprise an Mirantis ist die Zukunft von Swarm derzeit jedoch ungewiss. Sollte der neue Inhaber es wegen seiner bisherigen Vorliebe für Kubernetes einstellen, würde Swarm vielen Nutzern als einfacher und schneller Einstieg in die Container-Welt sicherlich fehlen.



Der Aufbau von zCX ähnelt Containern auf einer virtuellen Infrastruktur: Jede zCX hat hier einen eigenen IP-Adressraum und benötigt Routing und Firewall-Freischaltungen auf z/OS.
(Bild: IBM)

zCX lassen sich entweder als in sich geschlossener Cluster oder – die gängigere Variante – als ein oder mehrere Worker Nodes einsetzen, die in einem größeren Cluster besondere Aufgaben übernehmen.

Das kann beispielsweise die Kommunikation mit Anwendungen auf dem Host sein oder das Nutzen der Rechen- und Speicherkapazität des Hosts. Üblich ist auch eine gemischte Anwendung, bei der etwa ein Webserver für einen Anwendungsserver die Webschnittstelle einer nativen Hostanwendung bildet. So sind fertig paketierte Anwendungen für Linux schnell in die vorhandene Umgebung mit nativen z/OS-Anwendungen integrierbar.

Verschiedene Orchestrierungsansätze

Bei der Orchestrierung der Container stoßen zwei Ansätze aufeinander, wie sie eigentlich nicht unterschiedlicher sein könnten. Das Ziel ist, die Anwendung 24/7 zu betreiben. IBMs System Z hat seinen Namen von "Zero Downtime" und setzt dabei auf kostspielige Hardware, die auch in Clustern zusammenarbeitet. So stellt die Hardware sicher, dass die Anwendungen nicht ausfallen. Container auf Linux gehen einen anderen Weg: Sie gehen davon aus, dass die Hardware und der Container jederzeit ausfallen kann. Daher wird hier mit Redundanz und preiswerter Hardware gearbeitet, auf der redundant ausgelegte Software die Steuerung übernimmt. Sie ersetzt ausgefallene Container schnell mit neuen Instanzen auf wenn nötig anderer Hardware. Im Prinzip macht Kubernetes damit das, was auf der Z die Firmware und der Hypervisor machen, sie orchestrieren die Anwendungen und stellen den Betrieb sicher.

Mit den Methoden der Hochverfügbarkeit für Z lassen sich LPAR und Gäste in einem z/VM zwischen Standorten verschieben und sind so zu fast 100 % verfügbar oder nach lediglich kurzer Unterbrechung ohne Datenverluste wieder gestartet. Diese Garantie gilt – wenn zCX den Speicher eines Sysplex-Clusters an mehreren Standorten nutzt – dann auch inklusive Georedundanz für alle im zCX laufenden Container. Damit löst zCX das selbst für Swarm und Kubernetes bisher ungelöste Problem der Wiederherstellung an einem entfernten Standort, so wie es beispielsweise das Hochverfügbarkeitskompendium des BSI fordert.

Da mit Sysplex die zCX und alle darin laufenden Container (aus der Sicht der Orchestrierung) niemals ausfallen, wären also Swarm und zukünftig auch Kubernetes hier einer ihrer Hauptaufgaben beraubt. Alternativ verzichtet man auf die HA-Optionen von z/OS und nutzt die hochperformante Hardware nur als besonders schnelle Server. Das würde dem Host zwar nicht gerecht, könnte aber einiges an Lizenzkosten sparen. Allerdings ist zCX als Node in einem Cluster eher unüblich, es punktet mit der Einbindung in die Betriebsprozesse unter z/OS.

Ein oder mehrere Linux on Z – auf einem oder mehreren Hosts – mit Docker plus Swarm oder Kubernetes sind als Nodes in einem Cluster sehr viel besser geeignet und können auch die Master Nodes von Kubernetes betreiben.

Sicherheit und Mandantentrennung

Ein mit Kubernetes ungelöstes Problem ist das sichere Trennen von Mandanten innerhalb einer IaaS-Umgebung. Für Anwendungen, die einen besonders hohen Schutz vor Fremdeinwirkung durch Administratoren oder andere Anwendungen haben, sind ein geteilter Kubernetes-Cluster und IaaS auf einer geteilten physischen Infrastruktur daher nur bedingt geeignet. Deswegen wird in vielen Fällen für Anwendungen dann doch wieder eine in Hardware separierte Umgebung aufgebaut. Also pro

Anwendung, Fachbereich, Kunde oder Mandant jeweils eigene Server, Switches, Speicher – eigentlich genau das, was man mit Private Clouds und Cloud-Stack vermeiden wollte. Verdichtung und Sicherheit sind keine Freunde, wenn die logische Separierung nicht ausreicht.

Container auf Linux in Z und z/OS mit zCX haben hier eine Trumpfkarte, die sie ausspielen können. Eine Bedingung ist allerdings, dass sie in einer LAPR laufen, die gemäß dem Config-Guide für die EAL5+-Zertifizierung nach Common Critera für das Security Target der Separierung von logischen Partitionen aufgesetzt ist. Auch benötigt man für jeden Mandanten dann eine eigene, genauso konfigurierte LPAR mit entweder z/OS oder Linux on Z. Ein Host kann damit bis zu 80 Mandanten EAL5+-zertifiziert voneinander trennen. Dabei ist aber unbedingt zu beachten, dass CPU und I/O-Devices dann dediziert dieser LAPR zugewiesen sein müssen. Diese Konfiguration übertrifft bei Weitem die beispielsweise in der Verschlusssachenanordnung des Deutschen Staates geforderte EAL4+-Zertifizierung. Somit eignen sich beide Optionen auch für das Verarbeiten von Verschlusssachen mit erhöhten Einstufungen. Auch die in Linux on Z eingesetzten Linux-Distributionen sind vom BSI mit EAL4+-zertifiziert, wenn auch nicht unbedingt in ihren jeweils neuesten Versionen.

Ohne diese hohen Anforderungen oder wenn innerhalb eines Mandanten zum Beispiel Anwendungen voneinander zu separieren sind, kann auch die Separierung in Prozesse auf z/OS für zCX oder die Separierung in z/VM oder KVM auf Linux on Z zum Einsatz kommen. Für geringe Anforderungen an die Separierung eignen sich auch die Namespaces von Kubernetes. Diese lassen sich zwar mit Network Policies bis zur Mikrosegmentierung, Pod Security Policies, RBAC, Image-Scannern und so weiter noch erheblich verbessern, aber es bleibt bei einer logischen Separierung durch eine nicht zertifizierte Software. Der Artikel " **Virtualisierung: Tools zur Absicherung von Containern [3]** " liefert eine Übersicht von Kubernetes-Sicherheitsprodukten.

Fazit

IBMs Mainframes bieten eine sehr hohe Performance und Verfügbarkeit, die mit der Einführung der zCX und Linux on Z auch Containern zugutekommen kann. Performance, Sicherheit, Georedundanz und die EAL5+-zertifizierte Mandantentrennung sind starke Argumente für den Mainframe als Betriebsplattform auch für Container. Dies besonders, wenn mehrere Mandanten gemeinsam betrieben werden sollen. Verdichtung und agile Skalierung bei unterschiedlichen Sicherheitsanforderungen sind anders nur sehr schwer zu erreichen. Allerdings dürften die mit einem Wechsel der Plattform von x86-Servern auf einen Mainframe einhergehenden Umstellungen beim Betrieb auf viele Firmen und Behörden abschreckend wirken. Weiter sind besonders bei kompilierten Anwendungen einige Besonderheiten zu beachten.

Christoph Puppe ist Security Consultant bei SVA, Auditteamleiter für ISO 27001 nach Grundschutz, Mitautor des Grundschutzkompendiums und ehemaliger Penetrationstester. (**avr [4]**)

URL dieses Artikels:

`https://www.heise.de/-4617149`

Links in diesem Artikel:

[1] https://www.heise.de/ix/

[2]

[3] https://www.heise.de/hintergrund/Tools-zur-Absicherung-von-Containern-4340132.html

[4] mailto:avr@ix.de