OPENSTACK

Nutzung von Containern und OpenStack

Eine umfassende Übersicht

Contents

1. Einführung
2. I. Eine übergeordnete Ansicht von Containern in OpenStack
3. II. OpenStack Container Integrationspunkte
4. III. Fallstudien
5. IV. Fazit
6. V. Open-Source-Projektindex
7. VI. Autoren
8. Translators

Einführung

Stellen Sie sich vor, Sie müssten eine komplette private Cloud-Infrastruktur von Grund auf aufbauen. Sie haben ein begrenztes Budget, ein kleines, aber engagiertes Team und werden gebeten, ein Wunder zu vollbringen.

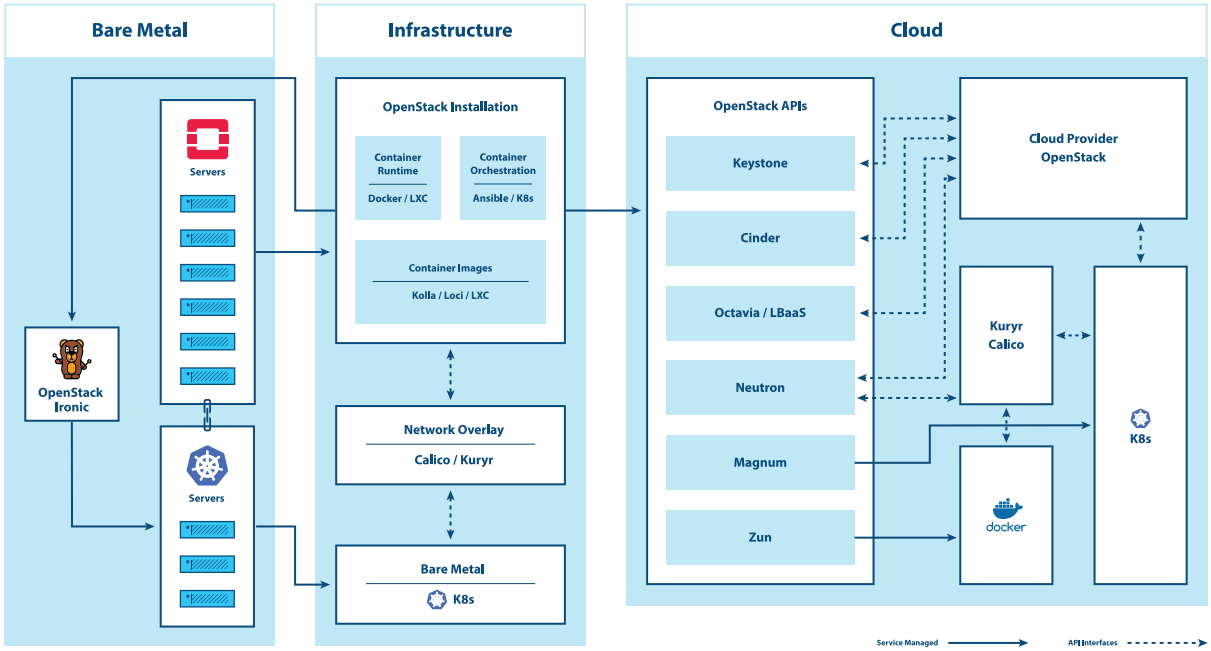
Vor ein paar Jahren würden Sie eine Infrastruktur mit Anwendungen erstellen, die auf virtuellen Maschinen laufen, mit einigen Bare-Metal-Maschinen für Legacy-Anwendungen. Mit der Entwicklung der Infrastruktur haben virtuelle Maschinen (VMs) ein höheres Maß an Effizienz und Agilität ermöglicht, aber VMs erfüllen die Anforderungen eines agilen Ansatzes für die Anwendungsbereitstellung nicht vollständig. Sie dienen weiterhin als Grundlage für die Ausführung vieler Anwendungen, aber Entwickler suchen zunehmend nach dem Trend zu Containern für die Entwicklung und den Einsatz modernster Anwendungen, da Container ein höheres Maß an Flexibilität und Effizienz bieten.

Container-Technologien wie Dockerund Kuberneteswerden zu den führenden Standards für den Bau von Container-Anwendungen. Sie helfen Organisationen von der Komplexität, die die Entwicklungsagilität einschränkt, zu befreien. Container-, Container-Infrastruktur- und Container-Deployment-Technologien haben sich als sehr mächtige Abstraktionen erwiesen, die auf verschiedene Anwendungsfälle angewendet werden können. Mit einem Unternehmen wie Kubernetes kann eine Organisation eine Cloud bereitstellen, die ausschließlich Container für die Anwendungsbereitstellung verwendet.

Bei einer hochmodernen Private Cloud geht es jedoch nicht nur um Container, und Container sind nicht für alle Workloads und Anwendungsfälle geeignet. Heutzutage müssen die meisten privaten Cloud-Infrastrukturen Bare-Metal-Maschinen zur Verwaltung der Infrastruktur, virtuelle Maschinen für Legacy-Anwendungen und Container für neuere Anwendungen umfassen. Die Fähigkeit, alle drei Ansätze zu unterstützen, zu verwalten und zu orchestrieren, ist der Schlüssel zur betrieblichen Effizienz.

OpenStackist derzeit die beste verfügbare Option für die Erstellung von privaten Clouds mit der Möglichkeit zur Verwaltung von Netzwerk-, Speicher- und Computerinfrastrukturen mit Unterstützung für virtuelle Maschinen, Bare-Metals und Container von einer Steuerungsebene aus. Obwohl Kubernetes wohl der beliebteste Container-Orchestrator ist und die Anwendungsbereitstellung geändert hat, hängt es von der Verfügbarkeit einer soliden Cloud-Infrastruktur ab, und OpenStack bietet die umfassendste Open Source-Infrastruktur für Hosting-Anwendungen. Die Multi-Tenant-Cloud-Infrastruktur von OpenStack eignet sich für Kubernetes mit mehreren Integrationspunkten, Bereitstellungslösungen und der Möglichkeit, mehrere Clouds miteinander zu verbinden.

In diesem Whitepaper werden wir untersuchen, wie Container innerhalb von OpenStack funktionieren, verschiedene Anwendungsfälle untersuchen und einen Überblick über Open Source-Projekte von OpenStack und anderswo geben, die dazu beitragen, Container zu einer Technologie zu machen, die einfach übernommen und genutzt werden kann.



I. Eine übergeordnete Ansicht von Containern in OpenStack

Es gibt drei Hauptszenarien, in denen sich Container und OpenStack kreuzen.

Das erste Szenario, die so genannten Infrastrukturcontainer, ermöglicht es Betreibern, Container so zu nutzen, dass die Bereitstellung, Verwaltung und der Betrieb der Cloud-Infrastruktur verbessert wird. In diesem Szenario werden Container auf einer Bare-Metal-Infrastruktur eingerichtet, und privilegierter Zugriff auf Hostressourcen ist zulässig. Dieser Zugriff ermöglicht es ihnen, Rechen-, Netzwerk- und Speicherressourcen direkt zu nutzen, die Containerlaufzeiten normalerweise vor Benutzern verbergen. Die Container isolieren die oft komplexen Abhängigkeiten, von denen jede Anwendung abhängt, während die Infrastrukturanwendungen die zugrunde liegenden Systemressourcen immer noch direkt verwalten und bearbeiten können. Wenn es an der Zeit ist, einen Dienst zu aktualisieren, kann das Upgrade ohne Änderungen in Abhängigkeiten gehandhabt werden und die gemeinsam lokalisierte Dienste zu stören.

Moderne Versionen von OpenStack haben dieses Infrastruktur-Containermodell übernommen, und es ist nun üblich, den gesamten Lebenszyklus einer OpenStack-Bereitstellung mit einer Kombination aus Orchestrierungswerkzeugen und containerisierten Diensten zu verwalten. Mit Infrastrukturcontainern können Container-Orchestrierungstechnologien zur Lösung zahlreicher Probleme eingesetzt werden, insbesondere im Hinblick auf die schnelle Iteration/Aktualisierung vorhandener Software einschließlich OpenStack. Durch das Ausführen von OpenStack in Containern können die Benutzer Tag 2-Herausforderungen lösen, z. B. neue Komponenten für

Services hinzufügen, Softwareversionen schnell aktualisieren und Updates schnell auf mehreren Computern und Rechenzentren ausführen. Dieser Ansatz bringt die Agilität von Containern auf das Problem der OpenStack-Bereitstellung und -Upgrades.

Das zweite Szenario befasst sich mit dem Hosting von containerisierten Anwendungs-Frameworks in der Cloud-Infrastruktur. Dazu können Container Orchestration Engines (COEs) wie Docker Swarm und Kubernetes oder Container-fokussierte Dienste mit geringerem Gewicht und serverbasierte Anwendungsprogrammierschnittstellen (APIs) gehören. Ob Bare-Metal oder VMs - die OpenStack-Community hat dafür gesorgt, dass Container-Anwendungen auf einem sicheren, mandanten-isolierten Cloud-Host bereitgestellt werden können. Dieses Szenario wird durch Treiber erleichtert, die es Projekten wie Kubernetes ermöglichen, OpenStack-APIs direkt für Speicher, Lastenausgleich und Identität zu nutzen. Es enthält auch APIs für die Bereitstellung von verwalteten Kubernetes-Clustern und Anwendungscontainern bei Bedarf. Mit diesen Funktionen können Entwicklungsteams neue containerisierte Anwendungen schreiben und Kubernetes-Cluster schnell auf OpenStack-Clouds bereitstellen. Es handelt sich um eine vollständige Application Lifecycle-Lösung, die ihnen die Ressourcen zur Verfügung stellt, die sie für die Entwicklung, das Testen und Debuggen ihres Codes benötigen, und eine robuste Automatisierung für die Bereitstellung ihrer Anwendungen in der Produktion.

Im letzten Szenario betrachten wir die Interaktionen zwischen unabhängigen OpenStack- und COE-Implementierungen und in diesem Artikel insbesondere Kubernetes-Cluster. Die Konsistenz und Interoperabilität der APIs in OpenStack- und Kubernetes-Clustern ist die wichtigste Quelle für den Erfolg dieses Szenarios. Zum Beispiel ist es möglich, dass Kubernetes direkt an von OpenStack Cindergehostete Volumes anfügt, OpenStack Keystoneals Autorisierungs- und Authentifizierungs-Backend verwendet oder eine Verbindung zu OpenStack Neutronals Netzwerk-Overlay mit OpenStack Kuryrherstellt . Umgekehrt ist es möglich, dass eine OpenStack-Cloud das gleiche Netzwerk-Overlay wie ein Kubernetes-Cluster mit Neutron-Treibern für Projekte wie Caliconutzt . Das dritte Szenario konzentriert sich weniger darauf, wie ein Cloud-Service gehostet wird (sei es Kubernetes oder OpenStack), sondern vielmehr darauf, wie unabhängige Dienste interagieren.

II. OpenStack Container Integrationspunkte

Bereitstellen der OpenStack-Infrastruktur für Container

Wie in der Einleitung erwähnt, hat sich die Bereitstellung und Verwaltung von OpenStack mit dem Aufkommen von Containern erheblich geändert, da Container neue Ansätze für die Verwaltung von Infrastrukturcode eröffnen. Frühere Managementstrategien erforderten entweder die Erstellung und Pflege von schwergewichtigen Golden Machine Images oder die Verwendung von empfindlichen Konfigurationsmanagementsystemen. Jeder Ansatz ist mit Komplexitäten und Einschränkungen verbunden. Zu dem Schwierigkeitsgrad kommt die Verwaltung einer Sammlung von Diensten hinzu, die alle ihre eigenen Abhängigkeiten benötigen, die sich von Release zu Release ändern. Ohne eine Form der Anwendungsisolierung wird das Lösen der Abhängigkeiten schwierig oder sogar unmöglich.

Mit Infrastrukturcontainern können neue OpenStack-Bereitstellungsprojekte ein Gleichgewicht zwischen beiden herstellen und gleichzeitig das Abhängigkeitsproblem elegant lösen. Mit leichtgewichtigen, unabhängigen, eigenständigen und in der Regel zustandslosen

Anwendungscontainern erhält ein Cloud-Operator enorme Flexibilität bei der Bereitstellung einer komplexen Steuerungsebene. In Kombination mit einer Container-Laufzeitumgebung und einer Orchestrierungs-Engine ermöglichen Infrastrukturcontainer die schnelle Bereitstellung, Wartung und Aktualisierung komplexer und hoch verfügbarer Infrastrukturen.

Beim Erstellen eines OpenStack-Clusters gibt es mehrere Dimensionen für die Auswahl von Bereitstellungstechnologien. Ein Operator kann Linux Containers(LXC) oder Docker für seine Basiscontainer auswählen, vordefinierte oder benutzerdefinierte Anwendungscontainer verwenden und entweder herkömmliche Konfigurationsmanagementsysteme für die Orchestrierung oder einen moderneren Ansatz wie Kubernetes auswählen. Tabelle 1 fasst die vorhandenen OpenStack-Bereitstellungsprojekte und ihre zugrunde liegenden Technologien zusammen.

Project	Container Type	Supported Containers	Project
OpenStack-Ansible	LXC	OSA LXC Containers	Ansible
Kolla-Ansible	Docker	Kolla Containers	Ansible
Triple-O	Docker	Kolla Containers	Ansible
OpenStack-Helm	Docker	Kolla Containers Loci Containers	Kubernetes and Helm

Table 1

Jedem dieser Bereitstellungssysteme liegen verschiedene Ansätze zum Erstellen einer Reihe von Containern für den OpenStack-Code und unterstützende Dienste zugrunde. Die OpenStack Ansible (OSA) - und Kolla-Projekte stellen ihre eigenen Projekt-gehosteten Build-Systeme zur Verfügung, während LOCI sich darauf konzentriert, Projekt-Anwendungscontainer zu erstellen, ohne ein spezielles Orchestrierungssystem im Auge zu behalten. Auf hohem Niveau sind die Unterschiede:

1. OSA ist insofern einzigartig, als es auf LXC-Containern auf niedrigerer Ebene basiert und ein benutzerdefiniertes Build-System zum Erstellen von LXC-Anwendungscontainern besitzt.
2. Das Kolla-Build-System erstellt Docker-Container, einen für jeden Service, sowie unterstützende Container für die Initialisierung und Verwaltung einer OpenStack-Bereitstellung. Kolla-Container sind in hohem Maße konfigurierbar, mit einer Auswahl an Basisbetriebssystem-, Quell- oder Paketinstallationen und einer Template-Engine für noch mehr Anpassung.
3. Die letzte Option zum Erstellen von OpenStack-Anwendungscontainern ist LOCI. LOCI erstellt auch Docker-Container und stellt für jedes Projekt einen Container bereit. LOCI

konzentriert sich darauf, kompakte und sichere Container für alle gängigen Distributionen schnell zu produzieren, mit der Erwartung, dass sie als Grundlage für das Deployment-System dienen.

Bare-Metal-Infrastruktur - OpenStack und Lösen des Bootstrap-Problems

Als Grundlage jeder Cloud gibt es ein Rechenzentrum von Bare-Metal-Servern, die die Infrastrukturdienste hosten. Selbst "Serverless Computing" führt Software in einer Cloud auf Hardware in einem Rechenzentrum aus. Das Problem der Bootstrap-Hardware-Infrastruktur ist ein kritisches Problem, das OpenStack-Software auf einzigartige Weise so ansprechen kann, dass Cloud-ähnliche Qualitäten dem Bare-Metal-Management verliehen werden.

OpenStack Ironicbietet Bare-Metal als Service. Als eigenständiger Dienst kann er Bare-Metal-Knoten erkennen, sie in einer Verwaltungsdatenbank katalogisieren und den gesamten Serverlebenszyklus einschließlich Registrierung, Bereitstellung, Wartung und Außerbetriebnahme verwalten. Als Treiber für OpenStack Novaund kombiniert mit allen OpenStack-Services bietet es einen leistungsstarken, Cloud-ähnlichen Service für die Verwaltung Ihrer gesamten Bare-Metal-Infrastruktur.

Dies wirft die Frage auf: Wie bootet man OpenStack-Dienste, um die Bare-Metal-Infrastruktur zu verwalten? Eine typische Lösung besteht darin, dieselben containerbasierten Installationswerkzeuge wie in den vorherigen Abschnitten beschrieben zu verwenden, um eine Startinstallation zu erstellen. Dieser Keim, der oft als "Undercloud" bezeichnet wird, kann verwendet werden, um das Management eines Bare-Metal-Clusters vollständig zu automatisieren, als wäre es eine virtualisierte Cloud.

Dies eröffnet die Möglichkeit, nicht nur die OpenStack-Virtualisierung in einer Bare-Metal-Cloud auszuführen, sondern auch reine Kubernetes-Installationen, die alle Vorteile der durch OpenStack-Dienste verfügbaren Identitäts-, Speicher-, Netzwerk- und anderen Cloud-APIs nutzen können .

Bereitstellung von containerbasierten Anwendungen auf OpenStack

Sowohl Infrastrukturcontainer als auch Bare-Metal-Infrastruktur sind wichtig, aber wenn die meisten Leute an Container denken, denken sie an Anwendungscontainer. Die Isolierung, Kapselung und Wartungsfreundlichkeit von Containern macht sie zu einer idealen Lösung für die Bereitstellung von Anwendungen. Container benötigen jedoch immer noch eine Host-Plattform, von der sie bedient werden können, sei es Bare-Metal, Public Cloud oder Private Cloud.

Kubernetes ist eine Plattform für die Bereitstellung von Anwendungen und funktioniert am besten mit Cloud-APIs, die die Bereitstellung kritischer Infrastrukturen wie permanenten Speicher, Lastenausgleich, Netzwerke und dynamische Zuweisung von Rechenknoten automatisieren können. OpenStack bietet Cloud-Infrastruktur, sei es als lokale Private Cloud oder über eine der verfügbaren öffentlichen oder verwalteten OpenStack-Clouds.

OpenStack war einer der ersten Upstream-Cloud-Anbieter für Kubernetes, mit einem aktiven Team von Entwicklern, die das " Kubernetes/Cloud Provider OpenStack" -Plugin pflegen. Dieses Plugin ermöglicht Kubernetes die Nutzung von Cinder Block Storage, Neutron und OctaviaLoad Balancers sowie die direkte Verwaltung von Rechenressourcen mit Nova. Die Verwendung des

Providers ist so einfach wie das Bereitstellen des Treibers für Ihre Kubernetes-Installation, das Festlegen eines Flags zum Laden des Treibers und das Bereitstellen der lokalen Benutzer-Cloud-Anmeldeinformationen.

Es gibt eine Reihe von Lösungen für die Installation von Kubernetes und anderen Anwendungsframeworks auf OpenStack. Eine der einfachsten Möglichkeiten, Container-Frameworks bereitzustellen , ist die Verwendung von Magnum, einem OpenStack-Projekt, das eine einfache API für die Bereitstellung vollständig verwalteter Cluster bereitstellt, die von mehreren Anwendungsplattformen unterstützt werden, einschließlich Kubernetes. Es ist ein Beispiel für ein Kubernetes-Bereitstellungssystem, das auf OpenStack-APIs und Cloud-Provider-Plug-Ins basiert. Zum Beispiel wird es zur Verwaltung von über 200 unabhängigen und föderierten Kubernetes-Installationen auf der CERN OpenStack-Cloud vor Ortsowie in Partner-Clouds verwendet. Wenn Ihnen die Magnum-API in Ihrer bevorzugten OpenStack-Cloud nicht zur Verfügung steht, können Sie alle anderen Kubernetes-Installationstools wie kubeadm, Kubernetes Anywhere, Cross-Cloudoder Kubespray verwenden, um Ihren Kubernetes-Cluster auf OpenStack zu installieren und zu verwalten . Da jeder Standard-Kubernetes verwendet, ist es einfach, die Cloud-Provider-Schnittstelle zu aktivieren, um Speicher- und Lastenausgleich zu nutzen.

Zun, ein weiteres OpenStack-Projekt, bietet eine leichtere Container-Service-API für die Verwaltung einzelner Container, ohne dass Server oder Cluster verwaltet werden müssen. Ein von OpenStack gehosteter Kubernetes-Cluster ist elastisch, da er dynamisch skaliert werden kann, indem Cloud-Ressourcen dem Cluster direkt über die Nova-API hinzugefügt oder daraus entfernt werden. Alternativ kann Kubernetes als Container-Backend für OpenStack Zun dienen und die Verwaltung der Pod-Infrastruktur an Zun übergeben. Es bietet eine leichtere und mandantenfähige Container-Service-API zum Ausführen von Containern, ohne dass direkt Server erstellt werden müssen. Die direkte Integration mit Neutron und Cinder ermöglicht die Vernetzung und Volumina einzelner Container.

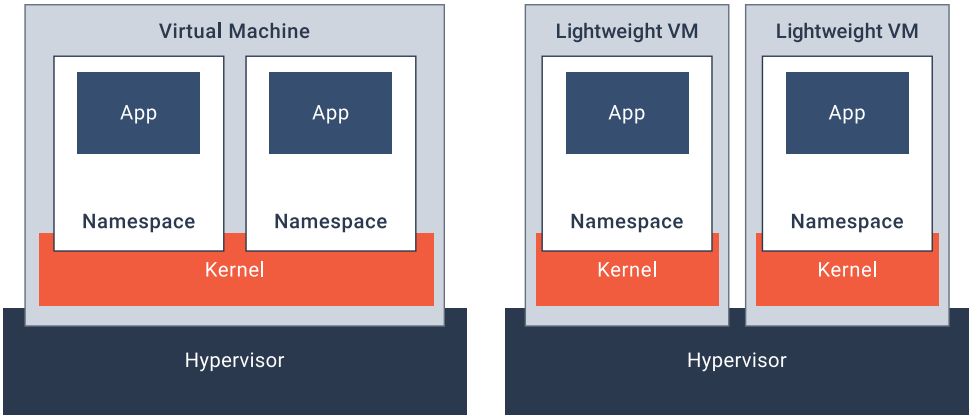
Schließlich bietet das Qinling-Projekt "Function as a Service", das darauf abzielt, eine Plattform zur Unterstützung serverloser Funktionen bereitzustellen, ähnlich wie Lambda, Azure Functions oder Google Cloud Functions. Es fasst die Verwaltung von Containern weiter zusammen und ermöglicht es Benutzern, die Entwicklung mit einer ereignisgesteuerten, serverfreien Computing-Erfahrung zu beschleunigen, die sich nach Bedarf skalieren lässt. Qinling unterstützt verschiedene Container-Orchestrierungs-Backends wie Kubernetes und Docker swarm, eine Vielzahl beliebter Speicher-Backends für Funktionspakete wie lokalen Speicher und OpenStack Swift.

Kata Container - Sichere Anwendungen durch Virtualisierung

Kata Containers, ein neues Open-Source-Projekt, ist eine neuartige Implementierung einer leichten virtuellen Maschine, die sich nahtlos in das Container-Ökosystem integriert. Kata Container sind so leicht und schnell wie Container und integrieren sich in die Container-Management-Schichten - einschließlich gängiger Orchestrierungs-Tools wie Docker und Kubernetes (k8s) - und bieten gleichzeitig die Sicherheitsvorteile von VMs. Kata Containers folgt dem Open Container Initiative (OCI)Standard, zu dem die OpenStack Foundation ein aktives Mitglied ist. Kata Containers wird bei der OpenStack Foundation gehostet, ist jedoch ein separates Projekt des OpenStack-Projekts mit eigener Governance und Community.

Der brancheninterne Wechsel zu Containern birgt einzigartige Herausforderungen bei der Sicherung von Benutzer-Workloads in Multi-Tenant-Umgebungen mit einer Mischung aus vertrauenswürdigen und nicht vertrauenswürdigen Workloads. Kata Containers verwendet hardwaregestützte Isolation als Grenze für jeden Container oder jede Sammlung von Containern in einem Pod. Dieser Ansatz behandelt die Sicherheitsanliegen eines gemeinsamen Kernels in einer herkömmlichen Containerarchitektur.

Kata Container eignen sich hervorragend sowohl für bedarfsgesteuerte, ereignisbasierte Bereitstellungen wie kontinuierliche Integration/kontinuierliche Bereitstellung als auch für länger laufende Webserveranwendungen. Kata ermöglicht auch einen einfacheren Übergang zu Containern aus herkömmlichen virtualisierten Umgebungen, da es Legacy-Gast-Kernel und Device-Pass-Through-Funktionen unterstützt. Kata Containers bietet verbesserte Sicherheit, Skalierbarkeit und höhere Ressourcenauslastung und führt gleichzeitig zu einem insgesamt vereinfachten Stack.



Containers in Cloud Today
(Shared kernel, isolation within namespace)

Kata Containers
A lightweight virtual machine isolates each container/pod and provides a separate kernel for each container/pod.

Side-by-Side-OpenStack- und Kubernetes-Integrationen

Einer der Hauptvorteile der Entscheidung für Open-Source-Plattformen liegt in der Stabilität der Schnittstellen bei Standardbereitstellungen dieser Plattformen. Sowohl die OpenStack Foundation als auch die Cloud Native Computing Foundation (CNCF) halten Interoperabilitätsstandards für OpenStack-Clouds und Kubernetes-Cluster ein und stellen sicher, dass Bibliotheken, Anwendungen und Treiber auf allen Plattformen funktionieren, unabhängig davon, wo sie eingesetzt werden. Dadurch ergeben sich Möglichkeiten für parallele Integrationen, sodass sowohl OpenStack als auch Kubernetes die Ressourcen der anderen nutzen können.

Die OpenStack Special Interest Group (SIG-OpenStack) in der Gemeinschaft Kubernetes unterhält das OpenStack-Plugin für Cloud-Anbieter. Neben der Cloud-Provider-Schnittstelle für die Ausführung von Kubernetes auf OpenStack werden auch mehrere Treiber verwaltet, mit denen Kubernetes einzelne OpenStack-Services nutzen kann. Diese Treiber umfassen:

- Zwei eigenständige Cinder-Treiber. Ein Flex Volume-Treiber verwendet ein exec-basiertes Modell als Schnittstelle zu Treibern und einen CSI-Treiber (Container Storage Interface), der eine Standardschnittstelle für Container-Orchestrierungssysteme verwendet, um beliebige Speichersysteme ihren Container-Workloads zur Verfügung zu stellen. Mit Unterstützung von über 70 Speichertreibern ermöglichen diese Treiber die Verbindung einer Vielzahl von kampferprobten proprietären und Open-Source-Speichergeräten über eine einzige Cinder-API.
- Eine Webhook-basierte Authentifizierungs- und Autorisierungsschnittstelle zu Keystone. Jeder Modus, Authentifizierung und Autorisierung, kann unabhängig voneinander konfiguriert werden. Obwohl eine Arbeit in Arbeit ist, unterstützt die Schnittstelle eine Soft-Mandantenfähigkeit, die Kubernetes RBAC mit OpenStack Keystone unterstützt.

Sowohl OpenStack als auch Kubernetes unterstützen hochdynamische Netzwerkmodelle, die von einer Vielzahl von Treibern unterstützt werden. Aufgrund dieser Standard-Netzwerkschnittstellen ist es einfach, eigenständige OpenStack- und Kubernetes-Cluster mit starken Netzwerkintegrationen zu erstellen. In OpenStack produziert das Kuryr-Projekt einen CNI-Treiber (Common Network Interface), der Neutron-Netzwerke an Docker und Kubernetes liefert. Auf der anderen Seite bieten Projekte wie Calico Neutron-Treiber an, die über Standard-Neutron-APIs direkten Zugriff auf beliebte Kubernetes-Netzwerk-Overlays bieten.

III. Fallstudien

Viele Mitglieder der OpenStack-Community stellen neuen Code für verschiedene OpenStack-Projekte zur Verfügung, die für Container relevant sind. Sie untersuchen die Auswirkungen und Vorteile von Containern und verwenden Container in der Produktion, um Herausforderungen zu lösen und neue Funktionen freizuschalten. In diesem Abschnitt werden einige der interessantesten Fallstudien vorgestellt.

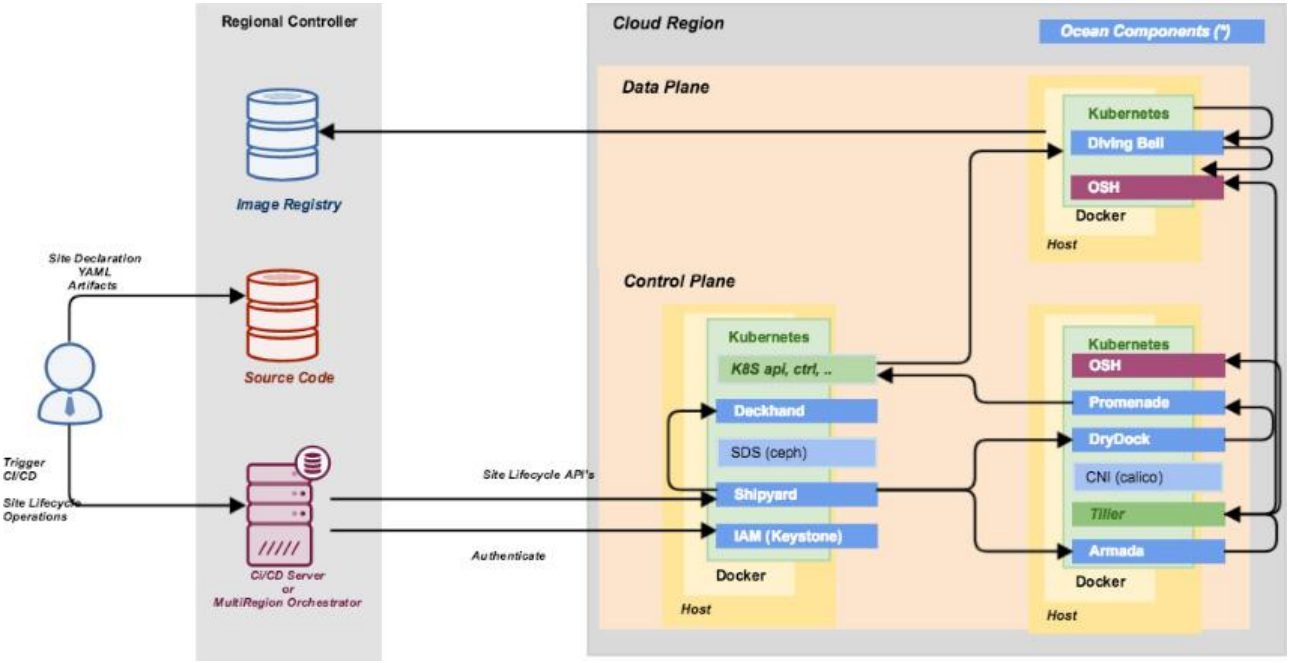
Select a Case Study

- AT&T
- Cern
- SK Telecom
- Superfluidität

AT&T

AT&T, eines der größten Telekommunikationsunternehmen der Welt, nutzt die Containertechnologie, um OpenStack selbst zu implementieren und zu verwalten. Dabei werden Infrastrukturcontainer verwendet, um Einfachheit und Effizienz zu erzeugen und ihre 5G-Infrastruktur auf Container-OpenStack aufzubauen.

Um ihre Ziele zu erreichen, nutzt AT&T das OpenStack-Helm-Projekt, um LOCI-basierte OpenStack-Bilder über einen Kubernetes-Cluster zu orchestrieren und dabei auch Kubernetes, Docker und die wichtigsten OpenStack-Dienste zu nutzen. Sie verwenden auch Bandit, Tempest, Patrole und viele andere OpenStack-Projekte. AT&T arbeitet auch in der Gemeinschaft zusammen, um eine Sammlung von Undercloud-Projekten namens Airshipvorzustellen, die Clouds von Bare-Metal bis zu Kubernetes in Produktionsqualität mit OpenStack-Workloads bereitstellen.



AT&T ist der Ansicht, dass die Containerisierung es ihnen ermöglicht, traditionelle Aktivitäten vom Typ Deployment weit nach links zu verlagern und sie mit CI/CD zu validieren. Kubernetes bietet zudem eine enorme Skalierbarkeit und Ausfallsicherheit sowie Hooks, mit denen OpenStack-Helm das Betriebsverhalten deklarativ konfigurieren, die Konfiguration injizieren und rollierende Upgrades und Updates durchführen kann, ohne die Belastung der Mandanten zu beeinträchtigen.

Die Nutzung der Containertechnologie für die Bereitstellung und Verwaltung von OpenStack sollte keine offensichtlichen Auswirkungen auf die Mandanten haben - mit der Ausnahme, dass sie eine widerstandsfähigere Plattform haben und Cloud-Funktionen häufiger und mit minimalen Unterbrechungen erhalten können. Die neuen Erfahrungen der AT&T-Betriebsteams werden ihre Bemühungen, die deklarative Konfiguration für eine Site zu definieren und die von Kubernetes-orientierte Automatisierung die Implementierungen selbst durchführen zu lassen, mehr und mehr verändern.

AT&T zielt darauf ab, diese Architektur zu nutzen, um die Funktionen des virtuellen Netzwerks zu unterstützen, die das Rückgrat seiner konsum- und geschäftsorientierten Produkte und Services bilden. Der anfängliche Anwendungsfall für AT&Ts Network Cloud mit Container wird die erstmalige Bereitstellung von VNFs für das entstehende 5G-Netzwerk sein. OpenStack war, ist und wird sich hervorragend für die VNF-fokussierten Cloud-Anwendungsfälle von AT&T

eignen. Die Containerisierung ist einfach eine Evolution, die es AT&T ermöglicht, ihre OpenStack-Infrastruktur auf eine zuverlässigere, schnellere und berührungslose Weise bereitzustellen, zu verwalten und zu skalieren.

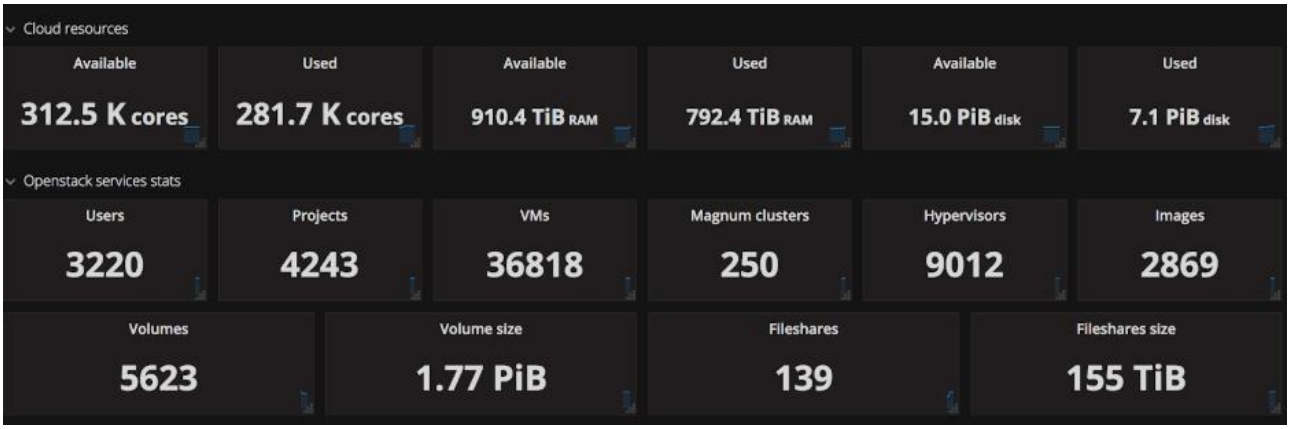
Operativ testet AT&T diesen Ansatz noch, hat sich jedoch verpflichtet, den 5G-Service noch in diesem Jahr in Betrieb zu nehmen. OpenStack- und Container-Technologie bilden das Rückgrat dieses für die Millionen von Nutzern von AT&T strategisch wichtigen Services. Die Bereitstellung ihres 5G-Dienstes demonstriert die Relevanz von OpenStack und Containern in einer massiv verteilten Produktionsumgebung.

Cern

CERN, die Europäische Organisation für Kernforschung, ermöglicht es Physikern und Ingenieuren, die fundamentale Struktur des Universums zu erforschen, indem sie die größten und komplexesten wissenschaftlichen Instrumente der Welt nutzen, um die grundlegenden Bestandteile der Materie - die fundamentalen Teilchen - zu untersuchen. Die CERN-Cloud bietet Physikern Rechenressourcen für das wissenschaftliche Rechnen, Analysieren von Daten aus dem Large Hadron Collider und andere Experimente.

CERN betreibt OpenStack seit 2013 in der Produktion und bietet nun Services für virtuelle Maschinen, Bare-Metal und Container in einer einzigen Cloud an. Container werden entweder auf virtuellen Maschinen oder auf Bare-Metal-Systemen ausgeführt, abhängig von den Anwendungsfällen, die alle über OpenStack Magnum bereitgestellt werden. Eine Auswahl verschiedener Containertechnologien ist verfügbar, darunter Kubernetes, Docker Swarm und DC/OS.

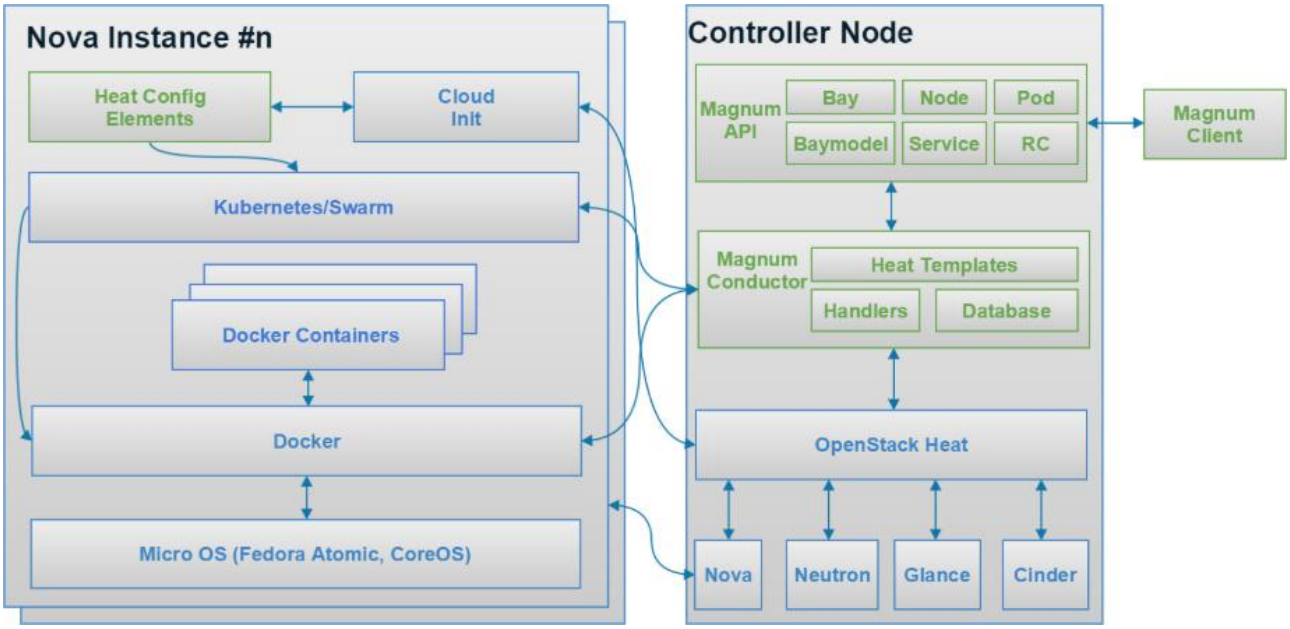
CERN betreibt derzeit 250 Container-Cluster, die über Magnum auf OpenStack bereitgestellt werden.



CERNs OpenStack-Cloud ermöglicht Benutzern den Self-Service-Zugriff, um eine konfigurierte Container-Engine mit einigen Befehlen oder über eine Web-GUI anzufordern. Dies ermöglicht eine schnelle Nutzung der Technologien und kann bei Bedarf auf 1000 Knoten skaliert werden. Best-Practice-Konfigurationen sind mit integrierter Überwachung und Integration in CERN-Speicher- und Authentifizierungsservices verfügbar.

Um diesen Ressourcenpool effizient zu verwalten und ihn ohne zusätzliche Arbeitsschritte zu skalieren, sind umfangreiche Managementprozesse und -tools erforderlich. Das Hinzufügen von Containern über Magnum über OpenStack ermöglichte es dem Service, die zuvor entwickelte Automatisierung zu nutzen, z. B. Hardware-Reparaturprozesse und konsistente Berechtigungsmodelle, während die schnelle Neuzuweisung von Ressourcen je nach Benutzeranforderungen unterstützt wurde.

Als öffentlich finanziertes Labor bieten Open-Source-Lösungen wie Kubernetes und OpenStack einen Rahmen, um mit anderen Organisationen zusammenzuarbeiten und den Gemeinschaften etwas zurückzugeben. CERN hat mit einer Reihe von Anbietern über das CERN-OpenLab-Frameworkwie Rackspace und Huawei zusammengearbeitet, um Clouds mit Funktionen wie Magnum und Federation im Maßstab zu liefern. Diese Erfahrungen werden auch durch OpenStack Special Interest Groups, mit anderen Wissenschaften wie dem Square Kilometre Array (SKA), öffentlichen Präsentationen wie Kubecon Europe und Blogs wie dem OpenStack in Productiongeteilt.



Am CERN laufen mehrere Workloads in Containern, die von Magnum bereitgestellt werden. Dazu gehören:

- Reana/Recast
  - Diese Tools bieten einen Rahmen für die Ausführung wiederverwendbarer Workflows in der Hochenergiephysik. Container bieten die Möglichkeit, die Analysesoftware und -daten in einer einzigen, leicht teilbaren Einheit zu packen und sowohl lokal als auch mithilfe externer Ressourcen problemlos zu skalieren. Die Arbeit ist als Kubernetes-Jobs geplant, die auf Yadage-Workflows basieren und Analyse- und Datenerhaltungsaktivitäten unterstützen.
- Spark as a Service
  - Vor kurzem wurde Kubernetes als Ressourcenmanager für Spark hinzugefügt. Spark kann Treiber und Executoren als Pods generieren und Kubernetes ist für die Planung und den Lebenszyklus verantwortlich. Ein Team in der IT-Abteilung von CERN entwickelt einen Service, mit dem Anwender bei Bedarf Kubernetes-Cluster

mit OpenStack Magnum erstellen und Spark auf Kubernetes bereitstellen können, um alle erforderlichen Integrationen mit den spezialisierten Dateisystemen und Datenquellen von CERN auf sichere Weise bereitzustellen. Benutzer mit wenigen Befehlen können effektiv eine Spark-Bereitstellung mit der gewünschten Größe erstellen, nur für die Zeit, die sie benötigen, und mit der Option, ihre Bereitstellung während der Ausführung nach oben oder unten zu skalieren.

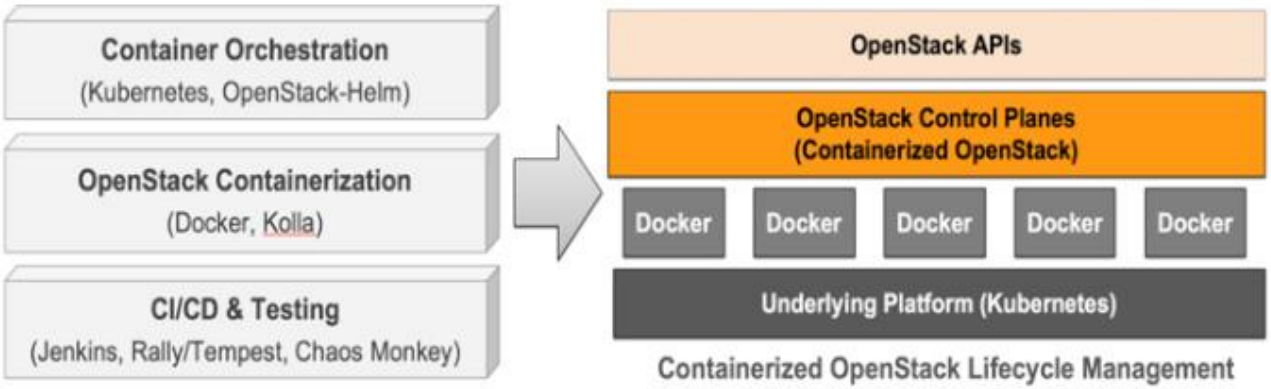
- LHC-Experiment-Detektor-Trigger-Simulation für LHC-Upgrade
  - Der LHC soll in den 2020er Jahren auf eine höhere Leuchtkraft gebracht werden, was erhebliche Verbesserungen in den Experiment-Trigger-Farmen erfordert, die die Kollisionen filtern. Große Kubernetes-Cluster wurden erstellt, um die verschiedenen Ansätze für das ATLAS-Experiment zu simulieren und das Design zu validieren, was zu einer Feinabstimmung der Kubernetes- und OpenStack-Komponenten führte.
- Gitlab Continuous Integration Runners
  - Mit Gitlab können Benutzer CI/CD-Jobs erstellen und diese auf gemeinsamen oder projektspezifischen Runnern ausführen. CERN-Benutzer können den CERN Container Service zum Testen und Erstellen von Software, zum Erstellen und Veröffentlichen von Container-Images und -Dokumentationen oder zum Einrichten komplexer Pipelines nutzen, um den gesamten Anwendungslebenszyklus einschließlich automatisierter Bereitstellungen in verschiedenen Umgebungen zu verwalten.
- Föderierte Kubernetes Compute Farmen mit externen Clouds
  - CERN verwendet Föderationen von Kubernetes-Clustern, um Multi-Cloud-Operationen zu unterstützen. Mehrere Cluster lassen sich nahtlos in Clouds unterschiedlicher Technologien integrieren, darunter AWS, GCE und OpenStack-Clouds wie CERN und die T-Systems Open Telekom Cloud, wie auf der Kubecon 2018 demonstriert wurde.

Die Integration virtueller Maschinen, Container-Engines und Bare-Metals in einem einzigen Framework bietet einfache Einblicke in die Nutzungsbilanzierung, den Besitz und das Kontingent. Manila-Speichertreiber für Kubernetes ermöglichen die transparente Bereitstellung von Dateifreigaben. Dies unterstützt sowohl die IT-Abteilung bei der Kapazitätsplanung als auch die Experiment-Ressourcen-Koordinatoren bei der Festlegung der Prioritäten für ihre Arbeitsgruppen. Ressourcenmanagement-Richtlinien wie die Neuzuordnung oder der Ablauf von Ressourcen bei der Abreise von Mitarbeitern werden in konsistenten Workflows behandelt.

SK Telecom

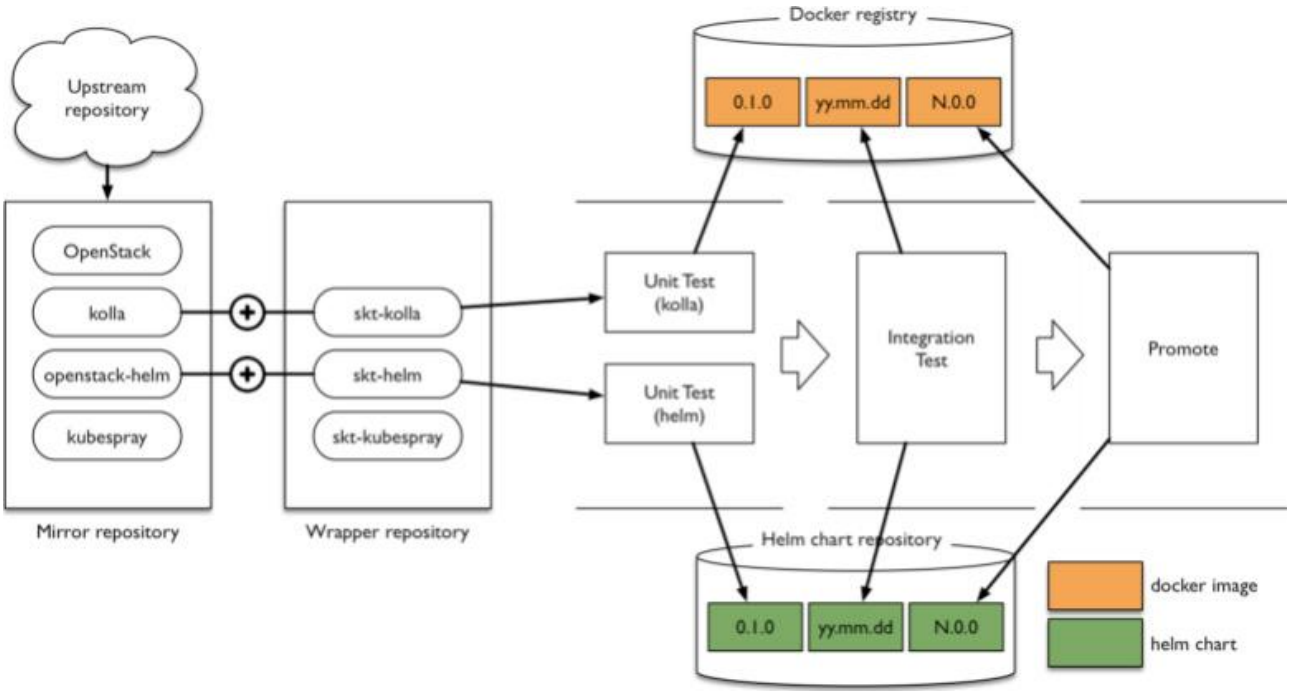
SK Telecom (SKT), South Korea’s largest telecommunications operator, has been exploring optimized approaches for deploying OpenStack on Kubernetes with the aim of putting core business functions on containerized OpenStack by the end of 2018. SKT leverages Kolla and Openstack-Helm. with deployments automated by Kubespray. SKT devotes nearly 100% of it’s development efforts to OpenStack-Helm, and works closely with AT&T to make OpenStack-Helm successful.

SKT hat auch andere Tools in ihren OpenStack zu Kubernetes Bemühungen integriert. Für Protokollierung, Überwachung und Alarme verwenden sie Prometheusund Elasticsearch, Fluent-Bitund Kibana, allesamt Standard-Referenzwerkzeuge in der OpenStack-Helm-Community. SKT kombiniert all diese in einer einzigen geschlossenen integrierten Lösung namens TACO: SKT All Container OpenStack.



SKT betont speziell eine automatisierte Continuous-Delivery (CI/CD) -Pipeline rund um den containerisierten Openstack auf Kubernetes. Das CI-System von SKT besteht aus Jenkins, Rally, Tempest, Docker Registry sowie Jira und Bitbucket. SKT entwickelte auch ein Open-Source-Tool namens Cookiemonster, ein "Chaos-Affe"-ähnliches Testwerkzeug für die Kubernetes-Implementierung, das Ausfallsicherheitstests für seine CI-Pipeline durchführt.

Mit jeder Änderung erstellt und testet SKT automatisch sowohl die OpenStack-Container als auch die Helm-Charts. Täglich installieren sie automatisch eine hochverfügbare OpenStack-Bereitstellung mit drei Steuerknoten und zwei Rechenknoten, führen 400 Testfälle von Tempest dagegen aus, um die Dienste zu validieren, und führen abschließend Ausfallsicherheitstests mit Cookiemonster und Rally durch. Das komplette CI-System ist im folgenden Diagramm dargestellt:



SKT automatisiert seine Einsätze mit Armada, einem Teilprojekt von Airship, das von AT&T als neues offenes Infrastrukturprojekt in der Gemeinde eingeführt wurde. SKT arbeitet in der Gemeinschaft zusammen, um Verbesserungen für das Projekt auf Basis ihrer Produktionsnutzung bereitzustellen.

In der Praxis hat SKT bereits eine große Anzahl von Vorteilen durch die Bereitstellung von OpenStack auf Kubernetes gesehen, darunter:

- Einfache und einfache Installation.
- Automatische Clusterheilung
- Eine Möglichkeit, OpenStack mit minimalen Auswirkungen auf die laufenden Dienste zu aktualisieren und zu aktualisieren.
- Schnelle Einführung fortgeschrittener Veröffentlichungsmethoden, einschließlich Blau-Grün-Bereitstellung, Canary-Versionen.
- Komplette automatisierte Verwaltung von Python-Abhängigkeiten durch Container-Isolation.
- Sicheres Geheimnis- und Konfigurationsmanagement.
- Schnelle und flexible Einführung von Cluster-Updates.

SKT testet immer noch den Ansatz, bewegt sich aber aktiv darauf hin, ihre OpenStack-Helm-Bereitstellungen in der Produktion auszuführen. Bis Ende dieses Jahres wird SKT mindestens drei Produktionscluster haben, von denen der vierte und der größte im Jahr 2019 online gehen werden. Diese Anwendungsfälle umfassen:

- Big Data-Plattform (geplant, um Q4 2018 live zu gehen)
- Eine virtuelle Desktop-Infrastruktur-Plattform (produktionsbereit ab Q4 2018)
- Eine allgemeine private Private Cloud (geplant für das 3. Quartal 2018)
- Eine Telco-Netzwerkinfrastruktur, die auf virtuellen Netzwerkfunktionen aufbaut (geplant für die Eröffnung im Jahr 2019)

SKT is also trying to improve automation on telecom infrastructure operation by utilizing containerized VNFs and leveraging containers' auto healing and fast scale-out features. In order to allow interaction between virtual machine based VNFs and containerized VNFs, Simplified Overlay Network Architecture(SONA), which is a virtual network solution for OpenStack, will support communication between VMs and containers. SONA uses the Kuryr project for integration of OpenStack and Kubernetes, and it optimizes network performance using software defined networking technologies.

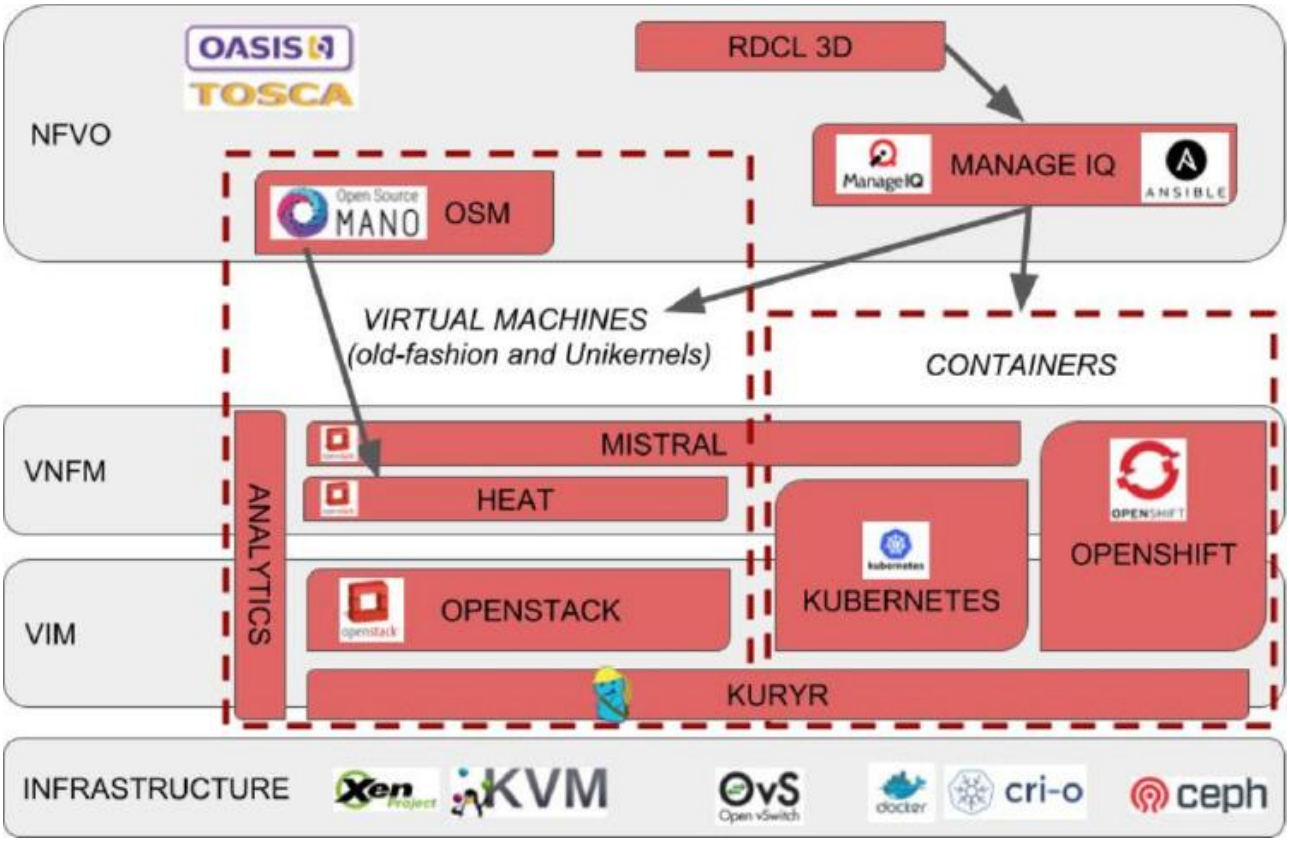
Insgesamt stellt SKT fest, dass Kubernetes dazu beiträgt, viele der komplexen Aspekte der Bereitstellung und des Betriebs von OpenStack zu lösen. Durch die Vereinfachung von OpenStack erhalten sie einen leistungsstarken Ansatz für die Bereitstellung innovativer Infrastruktur-Innovationen für die 5G-Ära. Durch die Fokussierung der Bemühungen um Openstack auf Kubernetes wurde die interne Fähigkeit zur Bewältigung der sich entwickelnden Verschiebung hin zu Microservices in Containern dramatisch erhöht und wurde zu einer kritischen Infrastruktur für die Bereitstellung von künstlicher Intelligenz, Internet der Dinge und maschinellem Lernen.

Superfluidität

Das Projekt Superfluiditybesteht aus 18 Partnern aus 12 europäischen Ländern. Es zielt darauf ab, die Fähigkeit zu verbessern, Dienste on-the-fly instanziieren, sie überall im Netzwerk (Kern, Aggregation, Rand) ausführen und sie transparent an verschiedenen Orten verschieben. SUPERFLUIDITY ist ein europäisches Forschungsprojekt (Horizon 2020), das versucht, die grundlegenden Infrastrukturblöcke für 5G-Netzwerke durch Nutzung und Erweiterung bekannter Open-Source-Projekte aufzubauen. SUPERFLUIDITY wird ein konvergentes Cloud-basiertes 5G-Konzept liefern, das innovative Anwendungsfälle im mobilen Umfeld ermöglicht, neue Geschäftsmodelle ermöglicht und Investitions- und Betriebskosten senkt.

Um diese Ziele zu erreichen, verschiebt sich das Projektkonsortium von älteren VM-basierten Anwendungen hin zu Cloud-Native-Container-Anwendungen. Kuryr dient als Brücke zwischen OpenStack Virtual Machines und Kubernetes und OpenShift Container-Services.

Das Projekt nutzt ManageIQals zentralen Netzwerkfunktions-Virtualisierungs-Orchestrator (NFVO), Ansible für Application Deployment und Lifecycle Management, OpenStack-Services wie Heat, Neutron und Octavia sowie Kubernetes über OpenShiftfür VMs und Container-Integration.



Durch die Verwendung von Ansible-Playbooks, die von der ManageIQ-Appliance ausgeführt werden, bietet SUPERFLUIDITY eine allgemeine Möglichkeit zum Bereitstellen von Anwendungen. Diese Anwendungen verwenden wiederum die Cloud-Orchestrierungsfunktionalität, die von OpenStack Heat-Vorlagen und OpenShift-Vorlagen bereitgestellt wird.

Das Konsortium setzt 5G-Cloud-Funkzugangsnetzwerke (CRAN) und mobile Edge-Computing-Komponenten (MEC) in Containern ein. Außerdem werden Anwendungen mit hohem Durchsatz wie Video-Streaming auf der verteilten Infrastruktur bereitgestellt.

Die Umstellung auf einen Cloud-nativen Ansatz für die Anwendungsbereitstellung ermöglicht schnelle und zuverlässige SUPERFLUIDITY-Installationen. Es ermöglicht einen reibungslosen Übergang von VM-basierten Anwendungen und Komponenten zu Containern, während gleichzeitig die Flexibilität beibehalten wird, VMs für bestimmte Anwendungen zu aktivieren. Beispiele für diese Anwendungen sind spezielle Sicherheitsmaßnahmen oder Netzwerkbeschleunigung, die für die Single-Route-Eingabe/Ausgabe-Virtualisierung (SRIOV) erforderlich sind.

Bei der Leistungsprüfung konnte SUPERFLUIDITY etwa 1000 Pods mit einer Geschwindigkeit von 22 Pods/Sekunde (mit von der Erstellung bis zum Lauf gemessener Zeit) starten. Diese bemerkenswerte Leistung wurde erzielt, indem OpenShift auf VMs ausgeführt wurde, die von OpenStack verwaltet wurden, wobei Kuryr als Pod-Netzwerktreiber fungierte, um Doppelkapselungsleistungstreffer zu vermeiden.

Select Another Case Study

IV. Fazit

In den letzten Jahren, als Container zu einem wichtigen Werkzeug für Entwickler und Organisationen wurden, hat OpenStack sein modulares Design und seine expansive Community genutzt, um Container-Technologien auf vielen Ebenen zu integrieren. Dies zeigt sich sowohl an den verschiedenen Organisationen, die Container und OpenStack in Produktion bringen, als auch an der Anzahl der Projekte, die neben den Containern arbeiten, um neue Funktionen bereitzustellen. Die OpenStack Foundation ist bestrebt sicherzustellen, dass aufkommende Technologien in OpenStack integriert und genutzt werden können, und Container sind ein wichtiges Beispiel für dieses Engagement.

Um mehr zu erfahren, besuchen Sie die Containers Landing Page, auf der Sie eine Kopie dieses Dokuments sowie Links zu Dutzenden von Videos zu den Integrationen von OpenStack und Containern finden. Kubernetes SIG-OpenStackverfügt über einen Slack-Kanal, eine Mailing-Liste und ein wöchentliches Meeting, wenn Sie direkt mit der Community interagieren, die Kubernetes und OpenStack-Integrationen erstellt.

V. Open-Source-Projektindex

Airship

Airship ist eine Sammlung von interoperablen und lose gekoppelten Open-Source-Tools, die auf Basis von Kubernetes als Anwendungsplattform eine automatische Cloud-Bereitstellung und -Verwaltung auf deklarative Weise bereitstellen.

Ansible

Ansible ist ein häufig verwendetes Orchestrierungswerkzeug zum Bereitstellen und Verwalten von OpenStack-Installationen.

Cinder

OpenStack Cinder bietet Blockspeicher als Service und stellt eine einzige API bereit, die von mehr als 70 verschiedenen möglichen Speichertreibern unterstützt wird.

Cloud-Anbieter OpenStack

Cloud Provider OpenStack ist die Implementierung der Kubernetes Cloud Provider Schnittstelle. Damit kann ein von OpenStack gehosteter Kubernetes-Cluster direkt auf Speicher- und Load-Balancer-Ressourcen in der OpenStack-Cloud zugreifen.

Calico

Calico ist ein Netzwerk-Overlay mit Treibern für Kubernetes und OpenStack, das nur L3-Routing bietet.

Cyborg

Cyborg ist ein OpenStack-Projekt, das ein allgemeines Management-Framework für Hardwarebeschleuniger einschließlich FPGA, GPU, ASIC und anderen bietet. Es wird daran gearbeitet, eine allgemeine Hardware-Schnittstelle zu Pods zu erstellen.

Docker

Docker ist ein Open-Source-Containervirtualisierungsframework, das zum Hosten von containerisierten Anwendungen verwendet wird.

Helm

Helm ist der offizielle Paketmanager für Kubernetes. Anwendungsbereitstellungen werden von Helm-Charts beschrieben, die automatisch auf einem Kubernetes-Cluster bereitgestellt und verwaltet werden können.

Ironic

Ironic ist der OpenStack-Bare-Metal-Service. Es kann entweder als eigenständiger Service oder als Treiber für OpenStack Nova ausgeführt werden und den gesamten Lebenszyklus von Bare-Metal-Systemen einschließlich Registrierung, Bereitstellung, Wartung und Außerbetriebnahme verwalten.

Loci

LOCI ist ein OpenStack-Projekt zum Erstellen von leichten, OCI-konformen Containern für OpenStack-Projekte.

LXC

LXC ist eine Low-Level-Container-Virtualisierungsschnittstelle, die die Isolierung von Linux-Kernel-Namespaces und andere Technologien nutzt, um isolierte Linux-Laufzeiten zu erstellen.

Kata-Container

Kata Container ist eine Standardimplementierung von leichtgewichtigen virtuellen Maschinen (VMs), die sich wie Container fühlen und ausführen, aber die Workload-Isolation und die Sicherheitsvorteile von VMs bieten.

Keystone

Keystone ist der OpenStack Identity Service, mit dem Benutzerkonten und Rolleninformationen primär für die OpenStack-Cloud-Umgebung authentifiziert und verwaltet werden, aber auch als Plugin für andere Umgebungen, einschließlich Kubernetes.

Kolla (Container)

Kolla (Container) ist ein OpenStack-Projekt zum Erstellen von Containern für jeden OpenStack-Service. Es umfasst ein ausgeklügeltes Build- und Templating-System und ist in der Lage, Container aus Quellen und Paketen auf einer Vielzahl von Host-Betriebssystemen zu erstellen.

Kolla Ansible

Kolla Ansible ist ein OpenStack-Projekt, das Ansible zum Bereitstellen und Verwalten einer vollständigen OpenStack-Installation mit Kolla-Containern verwendet.

Kubernetes

Kubernetes ist ein Container-Orchestrierungssystem, das robuste und hochverfügbare Anwendungen zusätzlich zur Cloud-Infrastruktur bereitstellt.

Kuryr

Kuryr ist ein OpenStack-Projekt, das Container-Laufzeiten, einschließlich Docker und Kubernetes, mit einem Neutron-Netzwerk-Overlay überlagert. Es soll die "Integrationsbrücke" für Container- und VM-Netzwerke sein.

Magnum

Magnum ist ein OpenStack-Projekt, das verwaltete Containerplattformen als Service anbietet, darunter Kubernetes, Docker Swarm, Mesos und DC/OS-Plattformen. Es ist in der Lage, mandantenunabhängige Anwendungsplattformen über eine einfache benutzerorientierte API zu erstellen.

Neutron

Neutron ist der softwarebasierte OpenStack-Netzwerkdienst, der eine einzige API für die Bereitstellung einer dynamischen Netzwerkinfrastruktur bietet, die von Dutzenden von Netzwerktreibern unterstützt wird.

OpenStack Ansible

OpenStack Ansible ist ein Projekt zum Erstellen von OpenStack-Diensten in LXC-Containern und zum Bereitstellen und Verwalten von OpenStack-Installationen innerhalb dieser Containerdienste.

OpenStack Helm

OpenStack Helm ist ein OpenStack-Projekt, das den Lebenszyklus von OpenStack und unterstützende Infrastruktur auf Kubernetes (z. B. Ceph und MariaDB) bereitstellt und verwaltet und einsatzbereite Bereitstellungen für eine Vielzahl von Anwendungsfällen von kleinen Edge-Implementierungen bis hin zu großen Zentralstellen bereitstellt. Nutzung des

Helm-Paketverwaltungssystems. OpenStack Helm unterstützt sowohl baremetales (Ironic) als auch virtuelles (Nova/KVM) Workload-Management und ist bildunabhängig und unterstützt sowohl LOCI- als auch Kolla-Container.

Qinling

Qinling ist ein OpenStack-Projekt zur Bereitstellung von Funktionen als Service. Qinling unterstützt verschiedene Container-Orchestrierungsplattformen wie Kubernetes und Docker Swarm sowie verschiedene Speicher-Backends für Funktionspakete wie den lokalen Dateispeicher, OpenStack Swift und S3.

Triple-O

TripleO ist ein Projekt zur Installation, Aktualisierung und zum Betrieb von OpenStack-Clouds mit OpenStacks Cloud-Services als Fundament für Nova, Iron, Neutron, Heat und Ansible zur Automatisierung des Cloud-Managements.

Zun

Zun ist der OpenStack Container-Dienst. Es zielt darauf ab, einen API-Dienst zum Ausführen von Anwendungscontainern bereitzustellen, ohne dass Server oder Cluster verwaltet werden müssen.

VI. Autoren

Mitglieder der OpenStack SIG-Kubernetes Community

- Jaesuk Ahn, SK Telecom
- Christian Berendt, Betacloud Solutions GmbH
- Anne Bertucio, OpenStack Foundation
- Pete Birley, AT&T
- Chris Hoge, OpenStack Foundation
- Lingxian Kong, Catalyst Cloud
- Hongbin Lu, Huawei
- Daniel Mellado, Red Hat, Inc.
- Allison Price, OpenStack Foundation
- David Rabel, B1 Systeme GmbH
- Sangho Shin, SK Telecom
- Davanum Srinivas, Huawei
- Luis Tomás, Red Hat, Inc.
- Sam Yaple, Verizon Digitale Mediendienste
- Mikhail Fedosin, Red Hat, Inc.
- Flavio Percoco, Red Hat, Inc.

Editor

- Brian E Whitaker, Zettabyte Inhalt LLC

Translators

- Frank Kloeker, <eumel@arcor.de>
- Robert Simai, <robert.simai@suse.com>