OPENSTACK

Memanfaatkan Kontainer dan OpenStack

Tinjauan Yang Komprehensif

Contents

1. Pengantar
2. I. A High Level View of Containers in OpenStack
3. II. OpenStack Container Integration Points
4. III. Case Studies
5. IV. Kesimpulan
6. V. Indeks Proyek Open Source
7. VI. Penulis
8. Translators

Pengantar

Imagine bahwa Anda ditugaskan untuk membangun infrastruktur cloud pribadi dari bawah ke atas. Anda memiliki anggaran terbatas, tim kecil tetapi berdedikasi, dan diminta untuk melakukan keajaiban.

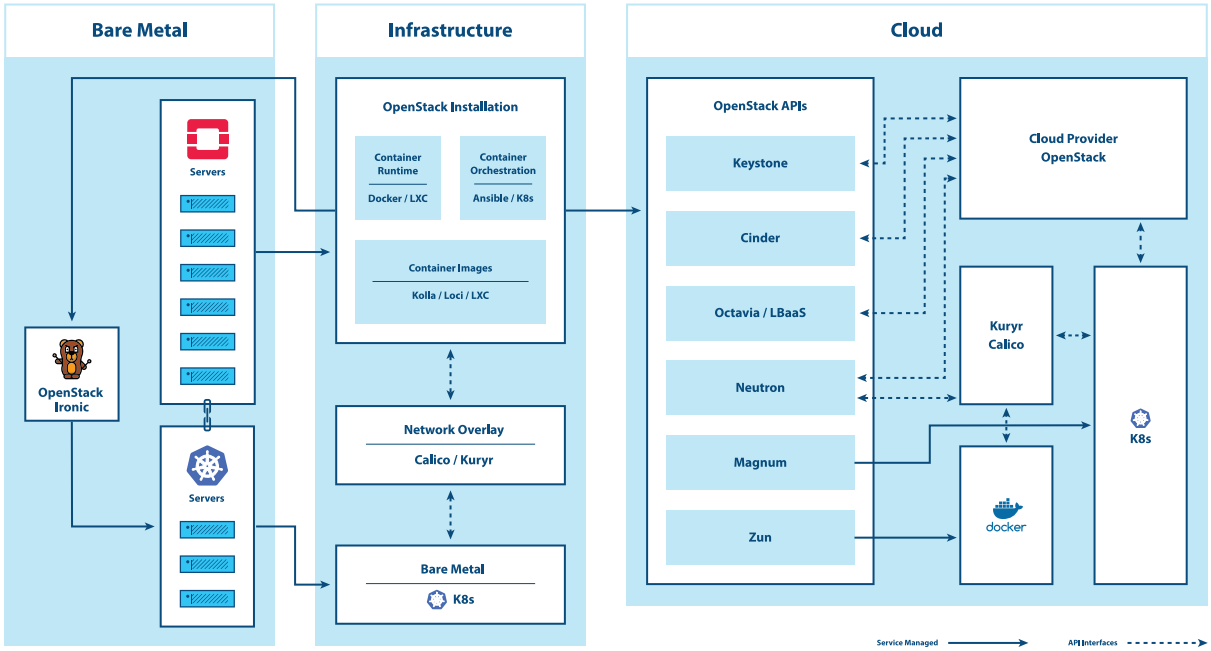
Beberapa tahun yang lalu, Anda membangun infrastruktur dengan aplikasi yang berjalan di mesin virtual, dengan beberapa mesin bare-metal untuk aplikasi warisan. Karena infrastruktur telah berevolusi, mesin virtual (VMs) memungkinkan tingkat efisiensi dan kelincahan yang lebih besar, tetapi VM saja tidak sepenuhnya memenuhi kebutuhan pendekatan gesit untuk penyebaran aplikasi. Mereka terus berfungsi sebagai landasan untuk menjalankan banyak aplikasi, tetapi semakin, pengembang mencari tren tren yang muncul untuk pengembangan dan penyebaran aplikasi terdepan karena kontainer menawarkan peningkatan tingkat kelincahan dan efisiensi.

Teknologi Container seperti Docker dan Kubernetes menjadi standar utama untuk membangun aplikasi kontainer. Mereka membantu organisasi bebas dari kompleksitas yang membatasi kelincahan pengembangan. Kontainer, infrastruktur kontainer, dan teknologi penyebaran kontainer telah membuktikan diri sebagai abstraksi yang sangat kuat yang dapat diterapkan pada sejumlah kasus penggunaan yang berbeda. Menggunakan sesuatu seperti Kubernetes, organisasi dapat mengirim awan yang hanya menggunakan kontainer untuk pengiriman aplikasi.

Namun, cloud pribadi terdepan tidak hanya tentang kontainer, dan kontainer tidak sesuai untuk semua workloads and use cases (kasus penggunaan). Saat ini, kebanyakan infrastruktur cloud swasta harus mencakup beberapa mesin bare-metal untuk mengelola infrastruktur, mesin virtual untuk aplikasi legacy, dan wadah untuk aplikasi yang lebih baru. Kemampuan untuk mendukung, mengelola dan mengatur ketiga pendekatan adalah kunci untuk efisiensi operasional.

OpenStacksaat ini adalah opsi terbaik yang tersedia untuk membangun awan pribadi, dengan kemampuan untuk mengelola jaringan, penyimpanan dan komputasi infrastruktur, dengan dukungan untuk mesin virtual, bare-metal, dan kontainer dari satu pesawat kontrol. Meskipun Kubernetes bisa dibilang orkestator wadah paling populer dan telah mengubah pengiriman aplikasi, itu tergantung pada ketersediaan infrastruktur cloud yang solid, dan OpenStack menawarkan infrastruktur open source paling komprehensif untuk aplikasi hosting. Infrastruktur multi-tenant cloud OpenStack adalah cocok secara alami untuk Kubernetes, dengan beberapa titik integrasi, solusi penyebaran, dan kemampuan untuk bergabung di banyak awan.

Dalam makalah ini, kami akan menyelidiki cara kerja kontainer di OpenStack, memeriksa berbagai kasus penggunaan, dan memberikan ikhtisar proyek sumber terbuka, dari OpenStack dan tempat lain, yang membantu membuat kontainer menjadi teknologi yang mudah diadopsi dan digunakan.



I. A High Level View of Containers in OpenStack

Ada tiga skenario utama di mana kontainer dan OpenStack berpotongan.

Skenario pertama, yang disebut kontainer infrastruktur, memungkinkan operator untuk memanfaatkan kontainer dengan cara yang meningkatkan penyebaran, pengelolaan, dan operasi infrastruktur awan. Dalam skenario ini, kontainer disiapkan pada infrastruktur bare-metal, dan diizinkan akses istimewa ke sumber daya host. Akses ini memungkinkan mereka untuk mengambil keuntungan langsung dari komputasi, jaringan, dan sumber daya penyimpanan yang runtime kontainer biasanya berusaha disembunyikan dari pengguna. Wadah mengisolasi set dependensi yang sering rumit yang bergantung pada setiap aplikasi, sementara masih memungkinkan aplikasi infrastruktur untuk secara langsung mengelola dan memanipulasi sumber daya sistem yang mendasarinya. Ketika saatnya tiba untuk meng-upgrade layanan, upgrade dapat ditangani tanpa perubahan dalam dependensi yang mengganggu layanan co-located.

Versi OpenStack modern telah merangkul model wadah infrastruktur ini, dan sekarang sudah normal untuk mengelola seluruh siklus hidup penyebaran OpenStack dengan kombinasi perkakas orkestrasi dan layanan kontainer. Kontainer infrastruktur memungkinkan operator untuk menggunakan teknologi orkestrasi kontainer untuk menyelesaikan banyak masalah, terutama di sekitar iterasi / peningkatan perangkat lunak yang ada dengan cepat termasuk OpenStack. Menjalankan OpenStack dalam kontainer membantu operator menyelesaikan tantangan 2 Hari, termasuk menambahkan komponen baru untuk layanan, meningkatkan versi perangkat lunak dengan cepat, dan pembaruan yang cepat di semua mesin dan pusat data. Pendekatan ini membawa kelincahan kontainer untuk masalah penyebaran dan peningkatan OpenStack.

Skenario kedua berkaitan dengan hosting kerangka aplikasi kontainer pada infrastruktur cloud. Ini dapat mencakup Container Orchestration Engine (COE) seperti Docker Swarm dan Kubernetes, atau layanan yang berfokus pada kontainer yang lebih ringan dan antarmuka pemrograman aplikasi tanpa server (API). Baik pada bare-metal atau VMs, komunitas OpenStack telah bekerja untuk memastikan bahwa mungkin untuk mengirim aplikasi yang kontainer aman, tenant-isolated cloud host. Skenario ini difasilitasi oleh driver yang memungkinkan proyek seperti Kubernetes untuk secara langsung memanfaatkan OpenStack API untuk penyimpanan, load balancing, dan identitas. Ini juga mencakup API untuk menyediakan kluster Kubernetes yang dikelola dan kontainer aplikasi sesuai permintaan. Dengan kemampuan ini, tim pengembangan dapat menulis aplikasi kontainer baru dan dengan cepat menyediakan kluster Kubernetes di awan OpenStack. Ini adalah solusi siklus hidup aplikasi lengkap yang memberi mereka sumber daya yang dibutuhkan untuk mengembangkan, menguji, dan men-debug kode mereka, dengan otomatisasi yang kuat untuk menyebarkan aplikasi mereka ke dalam produksi.

Dalam skenario terakhir, kami mempertimbangkan interaksi antara OpenStack independen dan penyebaran COE, dan dalam makalah ini terutama kluster Kubernetes. Konsistensi dan interoperabilitas API di seluruh OpenStack dan Kubernetes cluster adalah sumber utama keberhasilan untuk skenario ini. Misalnya, mungkin untuk Kubernetes langsung melampirkan ke OpenStack Cinder hosted volumes, gunakan OpenStack Keystone sebagai backend otorisasi dan otentikasi, atau terhubung ke OpenStack Neutron sebagai overlay jaringan dengan OpenStack Kuryr . Sebaliknya, ada kemungkinan awan OpenStack untuk berbagi overlay jaringan yang sama sebagai kluster Kubernetes dengan driver Neutron untuk proyek seperti Calico . Skenario ketiga kurang fokus pada bagaimana layanan cloud di-hosting (baik itu Kubernetes atau OpenStack), dan lebih banyak tentang bagaimana layanan independen berinteraksi.

II. OpenStack Container Integration Points

Deploying OpenStack Infrastructure on Containers

Seperti disebutkan dalam pendahuluan, penyebaran dan pengelolaan OpenStack telah berubah secara signifikan dengan munculnya kontainer, karena kontainer membuka pendekatan baru untuk mengelola kode infrastruktur. Strategi manajemen sebelumnya membutuhkan pembuatan dan pemeliharaan heavyweight golden machine images (image mesin emas kelas berat), atau menggunakan brittle state-maintaining configuration-management systems (sistem manajemen konfigurasi pemeliharaan yang rapuh). Setiap pendekatan hadir dengan kompleksitas dan pembatasan. Penambahan tingkat kesulitan adalah pengelolaan kumpulan layanan yang semuanya memerlukan dependensi mereka sendiri yang berubah dari rilis-ke-rilis. Tanpa beberapa bentuk isolasi aplikasi, penyelesaian untuk ketergantungan menjadi sulit jika bukan tidak mungkin.

Kontainer infrastruktur memungkinkan proyek penyebaran OpenStack baru untuk mencapai keseimbangan antara keduanya sementara secara elegan menyelesaikan masalah ketergantungan. Menggunakan kontainer aplikasi yang ringan, independen, mandiri, dan biasanya stateless, operator cloud mendapatkan fleksibilitas yang luar biasa ketika menerapkan complex control plane (bidang kontrol yang kompleks). Dikombinasikan dengan

runtime kontainer dan mesin orkestrasi, kontainer infrastruktur memungkinkan untuk menyebarkan, memelihara, dan meningkatkan infrastruktur yang kompleks dan sangat tersedia dengan cepat.

Dalam membangun cluster OpenStack, ada beberapa dimensi untuk memilih teknologi pengerahan. Operator dapat memilih Linux Containers(LXC) atau Docker untuk kontainer basis mereka, menggunakan kontainer aplikasi yang dibuat sebelumnya atau yang dibuat khusus, dan pilih sistem manajemen konfigurasi tradisional untuk orkestrasi atau pendekatan yang lebih modern seperti Kubernetes. Tabel 1 merangkum proyek pengerahan OpenStack yang ada dan teknologi yang mendasarinya.

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

Yang mendasari masing-masing sistem pengerahan ini adalah pendekatan yang berbeda untuk membangun sekumpulan kontainer untuk kode OpenStack dan layanan pendukung. Proyek OpenStack Ansible (OSA) dan Kolla menyediakan sistem pembangunan yang di-hosting proyek sendiri, sementara LOCI berfokus pada pembuatan kontainer aplikasi proyek, tanpa sistem orkestrasi tertentu. Pada tingkat tinggi, perbedaannya adalah:

1. OSA unik karena bergantung pada kontainer LXC tingkat rendah, dan memiliki sistem pembuatan kustom untuk membuat kontainer aplikasi LXC.
2. Sistem build Kolla menghasilkan kontainer Docker, satu untuk setiap layanan, bersama dengan kontainer pendukung untuk menginisialisasi dan mengelola pengerahan OpenStack. Kontainer Kolla sangat dapat dikonfigurasi, dengan pilihan sistem operasi dasar, instalasi sumber atau paket, dan mesin template untuk penyesuaian lebih lanjut.
3. Opsi terakhir untuk membangun kontainer aplikasi OpenStack adalah LOCI. LOCI juga membangun kontainer Docker, dan mengirimkan satu kontainer untuk setiap proyek. LOCI memfokuskan diri untuk memproduksi kontainer yang ringkas dan aman dengan cepat, untuk semua distribusi umum, dengan harapan bahwa mereka akan digunakan sebagai fondasi untuk dikembangkan oleh sistem pengerahan.

Bare-Metal Infrastructure - OpenStack dan Memecahkan Masalah Bootstrap

Di dasar setiap cloud, terdapat pusat data server-server bare-metal yang menjadi host layanan infrastruktur. Bahkan “serverless computing” menjalankan perangkat lunak pada cloud pada perangkat keras di pusat data. Masalah tentang bagaimana mem-bootstrap infrastruktur perangkat keras adalah masalah kritis yang perangkat lunak OpenStack secara unik memenuhi syarat untuk ditangani dengan cara yang memberikan kualitas seperti awan (cloud-like qualities) ke manajemen bare-metal.

OpenStack Ironic menyediakan bare-metal sebagai layanan. Sebagai layanan mandiri dapat menemukan node bare-metal, katalog mereka dalam database manajemen, dan mengelola siklus hidup seluruh server termasuk pendaftaran, penyediaan, pemeliharaan, dan dekomisioning. Ketika digunakan sebagai driver untuk OpenStack Nova dan dikombinasikan dengan rangkaian lengkap layanan OpenStack, itu memberikan layanan seperti awan yang kuat untuk mengelola seluruh infrastruktur bare-metal Anda.

Ini menimbulkan pertanyaan: Bagaimana cara satu bootstrap layanan OpenStack untuk mengelola infrastruktur bare-metal? Salah satu solusi yang umum adalah menggunakan alat instalasi berbasis kontainer yang sama seperti yang dijelaskan pada bagian sebelumnya untuk membuat instalasi benih. Benih ini, sering disebut 'undercloud', dapat digunakan untuk sepenuhnya mengotomatisasi pengelolaan klaster bare-metal seolah-olah itu adalah awan virtual.

Ini membuka peluang untuk tidak hanya menjalankan virtualisasi OpenStack pada cloud yang bare-metal, tetapi juga menjalankan instalasi bare-metal Kubernetes-onlyl yang dapat mengambil keuntungan penuh dari identity, penyimpanan, jaringan, dan API cloud lainnya yang tersedia melalui layanan OpenStack .

Delivering Container-Based Applications on OpenStack

Baik kontainer infrastruktur maupun infrastruktur bare-metal adalah penting, tetapi ketika kebanyakan orang berpikir tentang kontainer, mereka memikirkan kontainer aplikasi. Isolasi, enkapsulasi, dan kemudahan pemeliharaan yang ditawarkan oleh kontainer menjadikannya solusi ideal untuk pengiriman aplikasi. Namun, kontainer masih memerlukan platform host untuk melayani mereka dari, apakah bare-metal, cloud publik, atau cloud pribadi.

Kubernetes adalah platform untuk mengirimkan aplikasi, dan berfungsi paling baik dengan cloud-API yang dapat mengotomatiskan pengiriman infrastruktur penting seperti penyimpanan permanen, load-balancer, jaringan, dan alokasi dinamis dari node komputasi. OpenStack memberikan infrastruktur awan, baik sebagai awan pribadi on-prem atau melalui awan OpenStack publik atau terkelola yang tersedia.

OpenStack adalah salah satu penyedia awan upstream pertama untuk Kubernetes, dengan tim pengembang aktif yang mempertahankan plugin "Kubernetes/Cloud Provider OpenStack". Plugin ini memungkinkan Kubernetes untuk memanfaatkan penyimpanan blok Cinder, Neutron dan Load Balancers Octavia, dan manajemen langsung sumber daya komputasi dengan Nova. Menggunakan penyedia semudah mengirim driver ke instalasi Kubernetes Anda, menyetel bendera (flag) untuk memuat driver, dan menyediakan kredensial awan pengguna lokal Anda.

Ada sejumlah solusi untuk menginstal Kubernetes dan kerangka kerja aplikasi (application frameworks) lain di atas OpenStack. Salah satu cara termudah untuk memberikan kerangka kontainer adalah dengan menggunakan Magnum , proyek OpenStack yang menyediakan API sederhana untuk menerapkan kluster yang dikelola sepenuhnya didukung oleh pilihan beberapa platform aplikasi, termasuk Kubernetes. Ini adalah contoh sistem pengerahan Kubernetes yang bergantung pada OpenStack API dan plugin penyedia cloud. Misalnya, saat ini sedang digunakan untuk mengelola lebih dari 200 independen dan federasi Kubernetes installations on CERN's OpenStack on-site cloud, serta di awan mitra. Jika Anda tidak memiliki API Magnum tersedia untuk Anda di awan OpenStack pilihan Anda, Anda dapat menggunakan alat instalasi Kubernetes lainnya seperti kubeadm , Kubernetes Anywhere , Cross-Cloud , atau Kubespray , untuk menginstal dan mengelola kluster Kubernetes Anda di OpenStack. Karena masing-masing menggunakan Kubernetes standar, sangat mudah untuk mengaktifkan antarmuka penyedia cloud untuk memanfaatkan penyimpanan dan penyeimbangan beban.

Zun , proyek OpenStack lain, menawarkan API layanan kontainer yang lebih ringan untuk mengelola setiap kontainer tanpa kebutuhan untuk mengelola server atau kluster. Kluster Kubernetes yang dikelola OpenStack bersifat elastis karena dapat diubah secara dinamis dengan menambahkan atau menghapus sumber daya awan ke kluster secara langsung melalui API Nova. Atau, Kubernetes dapat berfungsi sebagai backend kontainer untuk OpenStack Zun, menyerahkan manajemen infrastruktur pod ke Zun. Ia menawarkan API layanan kontainer yang lebih ringan dan multi-tenant untuk menjalankan kontainer tanpa perlu membuat server secara langsung. Integrasi langsung dengan Neutron dan Cinder digunakan untuk menyediakan jaringan dan volume untuk masing-masing kontainer.

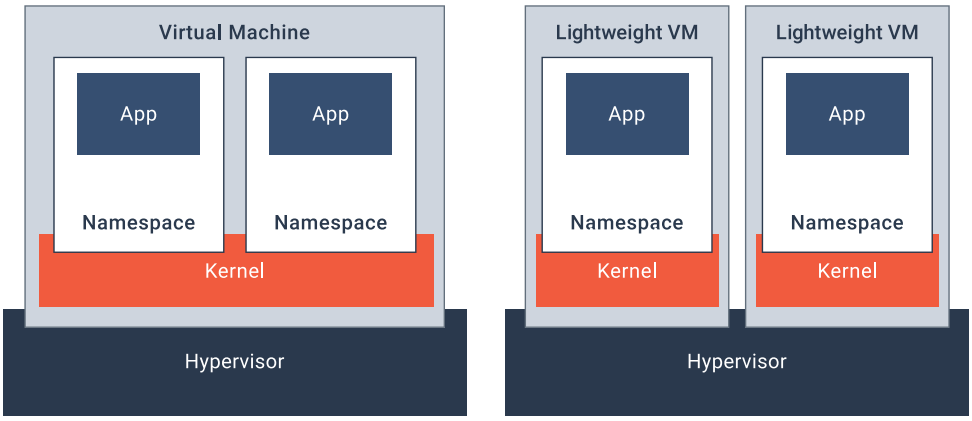
Akhirnya, proyek Qinling menawarkan "Function as a Service" yang bertujuan untuk menyediakan platform untuk mendukung fungsi serverless, mirip dengan Lambda, Azure Functions, atau Google Cloud Functions. Ini lebih lanjut abstrak pengelolaan kontainer, dan memungkinkan pengguna untuk mempercepat pengembangan dengan pengalaman komputasi berbasis-serverless, yang berskala sesuai permintaan. Qinling mendukung berbagai backends orkestrasi kontainer seperti Kubernetes dan Docker swarm, berbagai backend penyimpanan paket fungsi populer seperti penyimpanan lokal dan Swift OpenStack .

Kata Containers - Secure Applications through Virtualization

Kata Kontainer , proyek open source baru, adalah implementasi baru dari mesin virtual ringan yang terintegrasi dengan mulus (seamlessly) dalam ekosistem container. Kata Kontainer sangat ringan dan cepat seperti kontainer dan terintegrasi dengan lapisan pengelolaan kontainer - termasuk alat orkestrasi populer seperti Docker dan Kubernetes (k8s) - sementara juga memberikan keuntungan keamanan VMs. Kata Containers mematuhi standar Open Container Initiative (OCI) , yang mana OpenStack Foundation adalah anggota aktifnya. Kata Kontainer di-hosting di OpenStack Foundation, tetapi merupakan proyek terpisah dari proyek OpenStack dengan lembaga dan komunitasnya sendiri.

Pergeseran industri ke kontainer menghadirkan tantangan unik dalam mengamankan beban kerja pengguna dalam lingkungan multi-tenant dengan campuran beban kerja yang terpercaya dan tidak tepercaya. Kata Kontainer menggunakan isolasi yang didukung perangkat keras sebagai batas untuk setiap kontainer atau koleksi kontainer dalam pod. Pendekatan ini membahas masalah keamanan kernel bersama dalam arsitektur kontainer tradisional.

Kata Containers sangat cocok untuk penggunaan berdasarkan permintaan, berdasarkan peristiwa seperti integrasi berkelanjutan / pengiriman berkelanjutan, serta aplikasi server web yang berjalan lebih lama. Kata juga memungkinkan transisi yang lebih mudah ke penampung dari lingkungan virtualisasi tradisional, karena mendukung kernel dan perangkat lawas yang sudah lewat. Kata Kontainer memberikan peningkatan keamanan, skalabilitas dan pemanfaatan sumber daya yang lebih tinggi, sementara pada saat yang sama mengarah ke tumpukan yang disederhanakan secara keseluruhan.



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.

Integrasi OpenStack dan Kubernetes Side-by-Side

Salah satu manfaat utama memilih platform open source adalah stabilitas antarmuka di seluruh penerapan standar platform tersebut. Baik OpenStack Foundation dan Cloud Native Computing Foundation (CNCF) mempertahankan standar interoperabilitas untuk awan OpenStack dan kluster Kubernetes, menjamin bahwa pustaka, aplikasi, dan driver akan bekerja di semua platform terlepas dari di mana mereka dikerahkan. Ini menciptakan peluang untuk integrasi berdampingan, memungkinkan OpenStack dan Kubernetes untuk mengambil keuntungan dari sumber daya yang disediakan oleh yang lain.

OpenStack Special Interest Group (SIG-OpenStack) di komunitas Kubernetes memelihara plugin Cloud Provider OpenStack. Selain antarmuka penyedia cloud untuk menjalankan Kubernetes di OpenStack, itu juga mempertahankan beberapa driver yang memungkinkan Kubernetes untuk memanfaatkan layanan OpenStack individu. Driver ini termasuk:

- Dua driver Cinder yang berdiri sendiri. Driver Volume Flex menggunakan model berbasis-exec untuk berinteraksi dengan driver, dan driver Storage Storage Interface (CSI) yang menggunakan antarmuka standar untuk sistem orkestrasi kontainer untuk mengekspos sistem penyimpanan arbitrary ke beban kerja kontainer mereka. Dengan dukungan untuk lebih dari 70 driver penyimpanan, driver ini memungkinkan untuk

menghubungkan banyak perangkat penyimpanan sumber terbuka yang telah diuji coba melalui satu Cinder API.
- Autentikasi webhook-based dan antarmuka otorisasi berbasis web ke Keystone. Setiap mode, otentikasi dan otorisasi, dapat dikonfigurasi secara independen satu sama lain. Meskipun pekerjaan sedang berlangsung, antarmuka mendukung soft-multi-tenancy yang mendukung Kubernetes RBAC dengan OpenStack Keystone.

Baik OpenStack maupun Kubernetes mendukung model jaringan yang sangat dinamis yang didukung oleh berbagai driver. Karena antarmuka jaringan standar ini, mudah untuk membangun klaster OpenStack dan Kubernetes mandiri dengan integrasi jaringan yang kuat. Dalam OpenStack, proyek Kuryr menghasilkan driver Common Network Interface (CNI) yang mengirim jaringan Neutron ke Docker dan Kubernetes. Di sisi lain, ada proyek seperti Calico yang menawarkan driver Neutron, menyediakan akses langsung ke jaringan Kubernetes yang populer melalui API Neutron standar.

III. Case Studies

Banyak anggota komunitas OpenStack menyumbangkan kode baru ke berbagai proyek OpenStack yang relevan dengan penampung, mengevaluasi implikasi dan manfaat wadah, dan menggunakan kontainer dalam produksi untuk menyelesaikan tantangan dan membuka kemampuan baru. Bagian ini menyoroti beberapa studi kasus yang paling menarik.

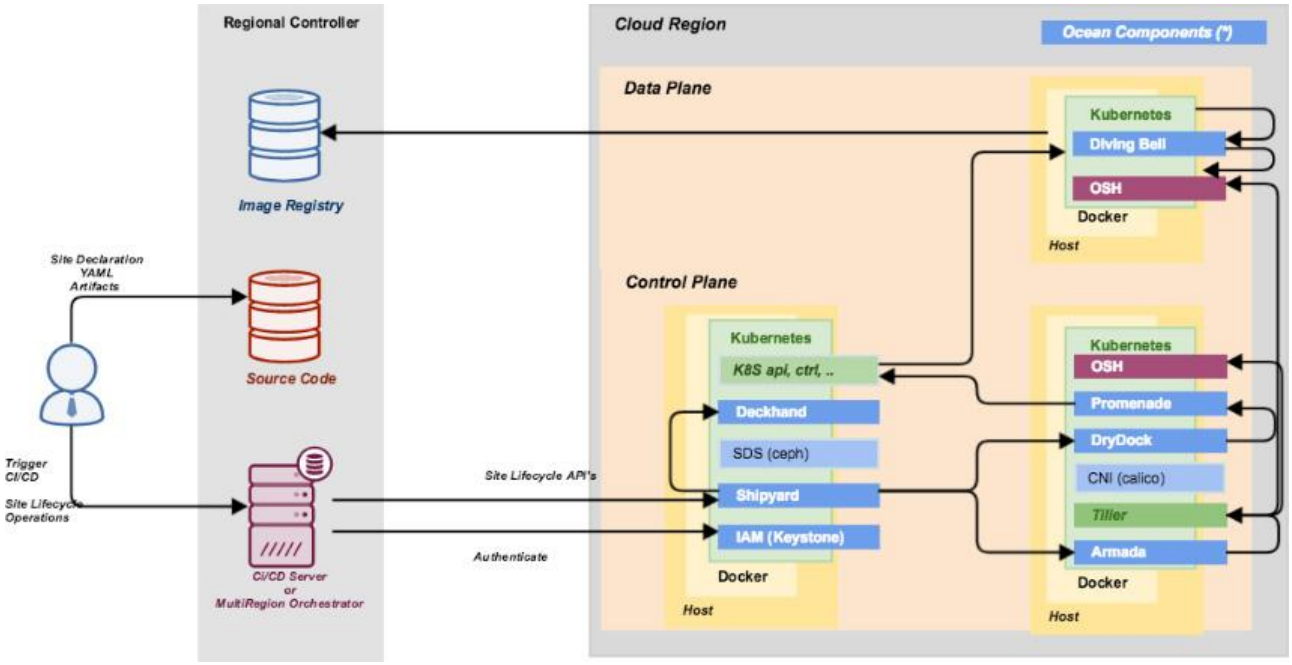
Select a Case Study

- AT&T
- Cern
- SK Telecom
- Superfluidity

AT&T

AT & T, salah satu perusahaan telekomunikasi terbesar di dunia, memanfaatkan teknologi kontainer untuk menyebarkan dan mengelola OpenStack itu sendiri, mengandalkan container infrastruktur untuk menghasilkan kesederhanaan dan efisiensi, dengan tujuan membangun infrastruktur 5G mereka di OpenStack yang terbungkus (containerized).

Untuk mencapai tujuan mereka, AT & T menggunakan proyek OpenStack-Helm untuk mengatur image OpenStack berbasis LOCI di seluruh kluster Kubernetes, juga memanfaatkan Kubernetes, Docker, dan layanan OpenStack inti. Mereka juga menggunakan Bandit, Tempest, Patrole, dan banyak proyek OpenStack lainnya. AT & T juga berkolaborasi dalam komunitas untuk memperkenalkan koleksi proyek undercloud yang disebut Airship , yang akan menyediakan awan dari bare-metal ke Kubernetes tingkat produksi yang menjalankan beban kerja OpenStack.



AT & T menemukan bahwa containerization memungkinkan mereka untuk mengalihkan aktivitas tipe pengerahan tradisional jauh ke kiri, dan memvalidasi mereka menggunakan CI/CD. Kubernetes juga menyediakan skalabilitas dan ketahanan besar, serta pengait untuk memungkinkan OpenStack-Helm mendeklarasikan perilaku operasional, menyetel konfigurasi, dan menyelesaikan peningkatan dan pembaruan secara bergulir tanpa memengaruhi beban kerja penyewa (tenant).

Memanfaatkan teknologi penampung untuk menerapkan dan mengelola OpenStack tidak akan memiliki banyak pengaruh nyata pada penyewa - dengan pengecualian bahwa mereka akan memiliki platform yang lebih tangguh, dan akan dapat memperoleh fitur cloud lebih sering dan dengan gangguan minimal. Pengoperasian AT&T tim pengalaman baru akan mengalihkan lebih banyak upaya mereka untuk menentukan konfigurasi deklaratif untuk situs, dan membiarkan otomatisasi yang berorientasi Kubernetes melakukan penerapan itu sendiri.

AT & T bertujuan untuk menggunakan arsitektur ini untuk memberdayakan fungsi-fungsi jaringan virtual yang membentuk tulang punggung dari produk dan layanan konsumer dan yang berfokus pada bisnis. Kasus penggunaan awal untuk AT & T Network Cloud yang terbendung akan menjadi penyebaran awal VNF untuk jaringan 5G yang muncul. OpenStack telah, dan akan sangat cocok untuk kasus-kasus penggunaan cloud VNF yang berfokus AT & T. Kontainer adalah evolusi yang memungkinkan AT & T untuk menyebarkan, mengelola, dan menskalakan infrastruktur OpenStack mereka dengan cara yang lebih dapat diandalkan, cepat, dan tanpa sentuhan.

Secara operasional, AT & T masih menguji pendekatan ini tetapi telah berkomitmen untuk mendapatkan layanan 5G ke dalam produksi sebelum akhir tahun. OpenStack dan teknologi kontainer akan membentuk tulang punggung dari layanan ini, yang secara strategis penting bagi jutaan pengguna AT & T. Menerapkan layanan 5G mereka akan menunjukkan relevansi OpenStack dan kontainer dalam lingkungan produksi yang didistribusikan secara besar-besaran.

Cern

CERN, Organisasi Eropa untuk Penelitian Nuklir, memungkinkan para fisikawan dan insinyur untuk menyelidiki struktur dasar alam semesta, menggunakan instrumen ilmiah terbesar dan paling rumit di dunia untuk mempelajari unsur dasar materi - partikel fundamental. CERN cloud menyediakan fisikawan dengan sumber komputasi untuk komputasi ilmiah, menganalisis data yang berasal dari Large Hadron Collider dan eksperimen lainnya.

CERN telah menjalankan OpenStack dalam produksi sejak 2013 dan sekarang menyediakan layanan untuk mesin virtual, bare-metal, dan kontainer dalam satu cloud. Container berjalan di mesin virtual atau bare-metal tergantung pada use case, semua disediakan melalui OpenStack Magnum. Pilihan teknologi kontainer yang bermacam macam tersedia termasuk Kubernetes, Docker Swarm dan DC/OS.

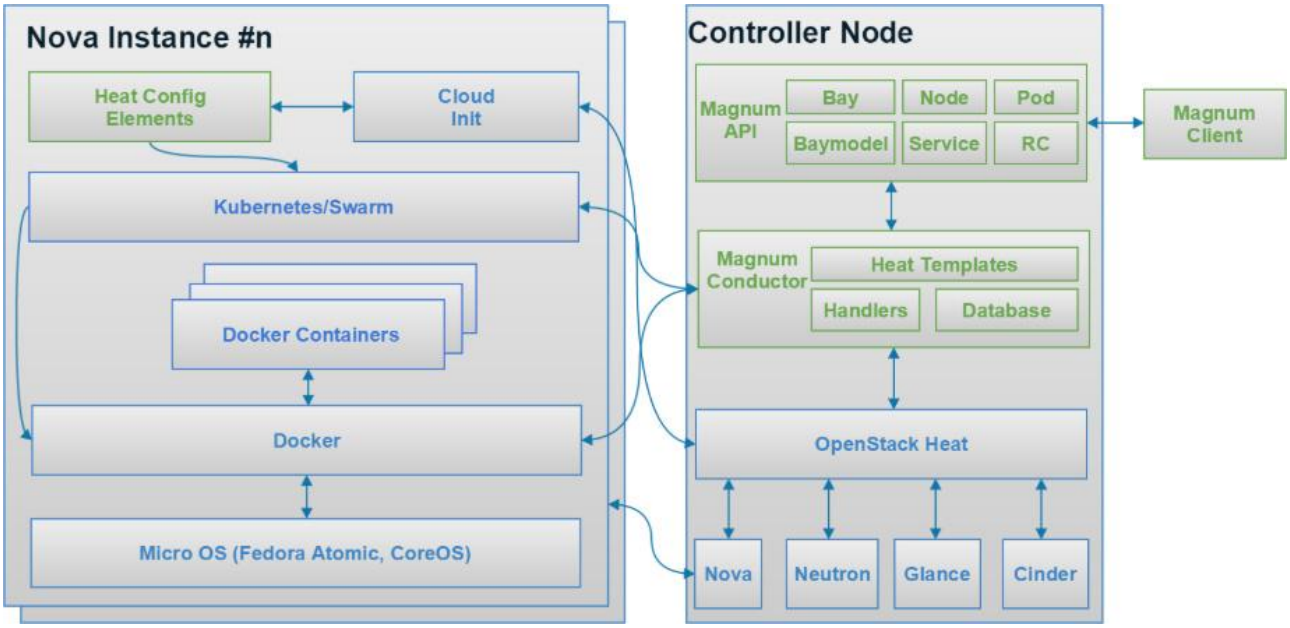
CERN saat ini menjalankan 250 kelompok kontainer yang disediakan melalui Magnum di atas OpenStack.



OpenStack cloud CERN memberi pengguna akses swalayan untuk meminta mesin container yang dikonfigurasi dengan beberapa perintah atau melalui GUI web. Hal ini memungkinkan pemanfaatan teknologi secara cepat dan dapat menskalakan hingga 1000 node jika diperlukan. Konfigurasi praktik terbaik tersedia dengan pemantauan dan integrasi yang terintegrasi ke dalam layanan penyimpanan dan otentikasi CERN.

Menjalankan kumpulan sumber daya ini secara efisien, menskalakannya tanpa perlu operasi ekstra, tenaga kerja membutuhkan proses dan alat manajemen yang konsisten. Menambahkan container melalui Magnum di atas OpenStack memungkinkan layanan untuk menggunakan otomatisasi yang sebelumnya dikembangkan, seperti proses perbaikan perangkat keras dan model otorisasi yang konsisten sambil mendukung realokasi sumber daya dengan cepat tergantung pada kebutuhan pengguna.

Sebagai laboratorium yang didanai publik, solusi sumber terbuka seperti Kubernetes dan OpenStack menyediakan kerangka kerja untuk berkolaborasi dengan organisasi lain dan memberikan kembali kepada masyarakat. CERN telah bekerja dengan sejumlah vendor melalui CERN openlab framework, seperti Rackspace dan Huawei, untuk menyediakan awan dalam skala dengan fungsi seperti Magnum dan federasi. Pengalaman ini juga dibagikan melalui OpenStack Special Interest Groups, dengan ilmu lain seperti Square Kilometer Array (SKA), presentasi publik seperti Kubecon Europe dan blog seperti OpenStack in Production.



Di CERN, beberapa beban kerja dijalankan dalam kontainer yang disediakan oleh Magnum, ini termasuk:

- Reana/Recast
  - Alat-alat ini menyediakan kerangka kerja untuk mengeksekusi alur kerja yang dapat digunakan kembali di High Energy Physics. Container menawarkan kemampuan untuk mengemas perangkat lunak dan data analisis dalam satu unit yang mudah dibagikan serta penskalaan yang mudah di luar lokasi dan menggunakan sumber daya eksternal. Pekerjaan dijadwalkan sebagai pekerjaan Kubernetes berdasarkan Yadage Workflow yang mendukung analisis dan aktivitas pelestarian data.
- Spark as a Service
  - Baru-baru ini, Kubernetes ditambahkan sebagai pengelola sumber daya untuk Spark. Spark dapat menelurkan driver dan eksekutor sebagai pod dan Kubernetes bertanggung jawab atas penjadwalan dan siklus hidup. Sebuah tim di departemen TI CERN sedang mengembangkan layanan di mana pengguna dapat membuat kluster Kubernetes sesuai permintaan dengan OpenStack Magnum dan menyebarkan Spark pada Kubernetes, menyediakan semua integrasi yang diperlukan dengan sistem file dan sumber data khusus CERN dengan cara yang aman. Pengguna dengan beberapa perintah dapat secara efektif membuat penyebaran Spark dengan ukuran yang diinginkan, hanya untuk saat mereka membutuhkannya dan dengan opsi untuk meningkatkan atau menurunkan penyebaran mereka saat berjalan.
- Simulasi pemicu detektor percobaan LHC untuk upgrade LHC
  - LHC akan ditingkatkan ke luminositas yang lebih tinggi selama 2020-an yang membutuhkan peningkatan signifikan dalam percobaan pemicu peternakan (trigger farm) yang menyaring tumbukan (collision). Kluster Kubernetes skala besar telah dibuat untuk mensimulasikan pendekatan yang berbeda untuk eksperimen ATLAS dan memvalidasi desain, menghasilkan beberapa penyempurnaan komponen Kubernetes dan OpenStack.

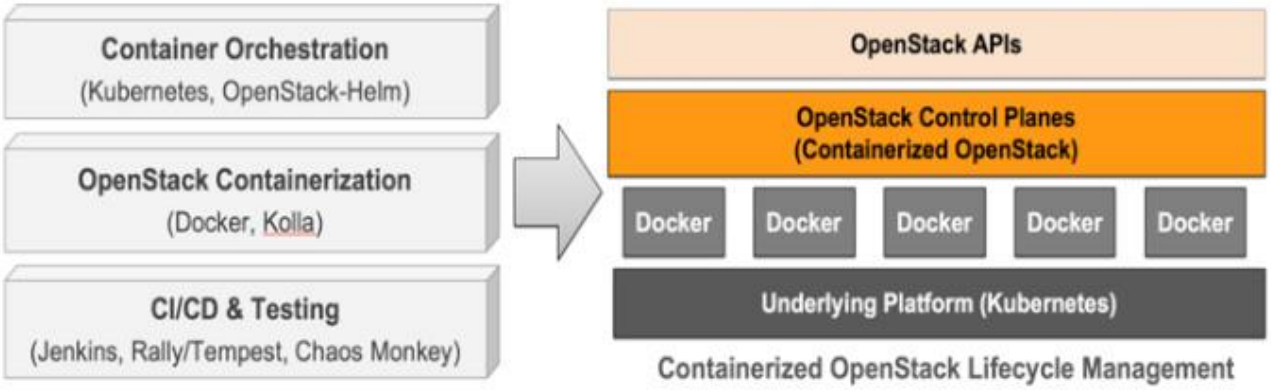
- Gitlab Continuous Integration Runners
    - Gitlab memungkinkan pengguna untuk membangun pekerjaan CI/CD dan mengeksekusi mereka pada shared or project specific runneri. Pengguna CERN dapat memanfaatkan CERN Container Service untuk menguji dan membangun perangkat lunak, membuat dan mempublikasikan image dan dokumentasi container atau mengatur pipeline kompleks yang mengelola siklus hidup aplikasi lengkap, termasuk penerapan otomatis ke berbagai lingkungan.
- Federated Kubernetes compute farm dengan awan eksternal
    - CERN menggunakan federasi kluster Kubernetes untuk mendukung operasi multi-cloud. Beberapa gugus dapat diintegrasikan secara mulus di awan berbagai teknologi, termasuk AWS, GCE, dan awan OpenStack seperti CERN dan T-Systems Open Telekom Cloud. as demonstrated at Kubecon 2018.

Mengintegrasikan mesin virtual, mesin kontainer, dan bare-metal di bawah satu kerangka kerja memberikan pandangan yang mudah tentang penggunaan akuntansi, kepemilikan, dan kuota. Driver penyimpanan Manila untuk Kubernetes memungkinkan penyediaan file secara transparan. Ini mendukung baik departemen TI dalam perencanaan kapasitas dan koordinator sumber daya percobaan dalam menentukan prioritas untuk kelompok kerja mereka. Kebijakan manajemen sumber daya seperti penugasan kembali (reassignment) atau kadaluwarsa sumber daya (expiry of resource) pada kepergian staf ditangani dalam alur kerja yang konsisten.

SK Telecom

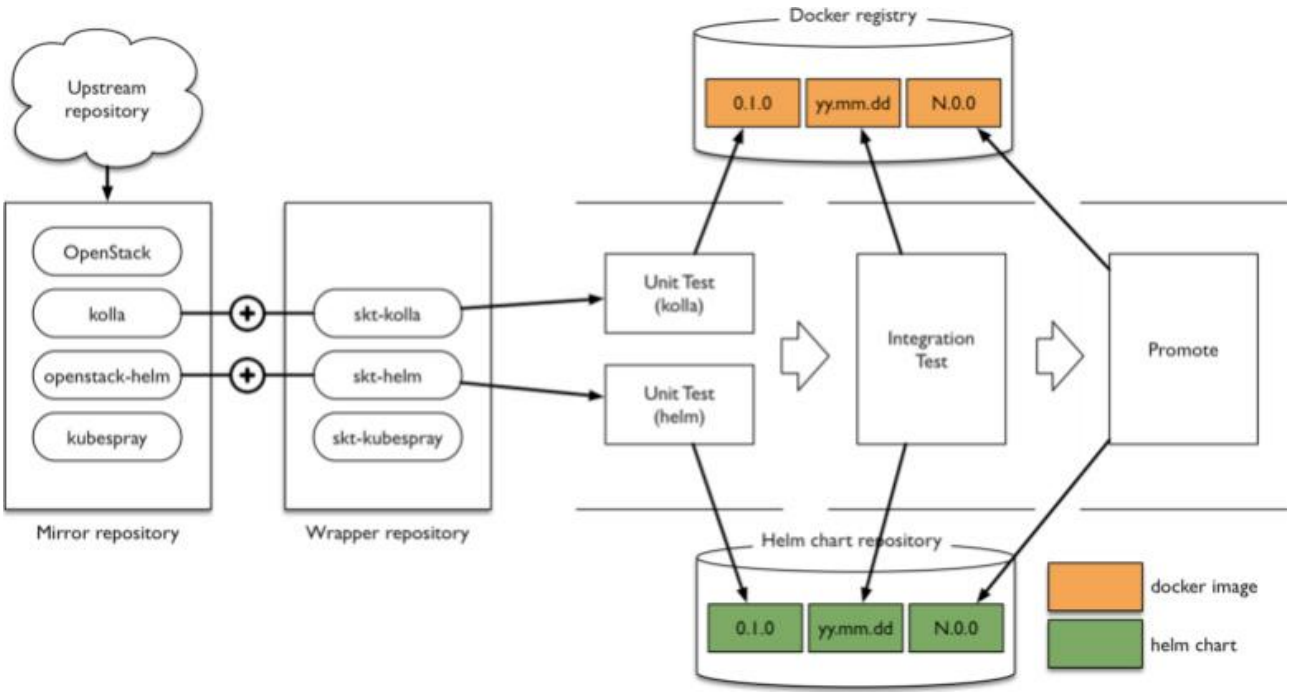
SK Telecom (SKT), South Korea's largest telecommunications operator, has been exploring optimized approaches for deploying OpenStack on Kubernetes with the aim of putting core business functions on containerized OpenStack by the end of 2018. SKT leverages Kolla and Openstack-Helm. with deployments automated 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 juga memasukkan alat-alat lain ke OpenStack mereka dalam upaya Kubernetes. Untuk pencatatan, pemantauan, dan alarm, mereka menggunakan Prometheusand Elasticsearch, Fluent-bit, and Kibana, semuanya merupakan alat referensi default dalam komunitas OpenStack-Helm. SKT menggabungkan semua ini ke dalam solusi terintegrasi tunggal yang disebut TACO: SKT All Container OpenStack.



SKT secara khusus menekankan continuous integration/continuous delivery (CI / CD) secara otomatis di sekitar Openstack di Kubernetes. Sistem CI SKT terdiri dari Jenkins, Rally, Tempest, Docker Registry, serta Jira dan Bitbucket. SKT juga mengembangkan alat open source yang disebut Cookiemonster, chaos-monkey seperti alat uji ketahanan untuk penyebaran Kubernetes yang melakukan tes ketahanan untuk pipa CI mereka.

Dengan setiap perubahan, SKT secara otomatis membuat dan menguji kontainer OpenStack dan grafik Helm. Setiap hari, mereka secara otomatis menginstal penyebaran OpenStack yang sangat tersedia dengan tiga node kontrol dan dua node komputasi, menjalankan 400 uji kasus dari Tempest melawannya untuk memvalidasi layanan, dan akhirnya menjalankan pengujian ketahanan dengan Cookiemonster dan Rally. Sistem CI lengkap diilustrasikan dalam diagram berikut:



SKT mengotomatiskan pengerahannya dengan Armada, a sub-project of Airship, yang diperkenalkan di masyarakat sebagai proyek infrastruktur terbuka baru oleh AT & T. SKT berkolaborasi dalam komunitas untuk memberikan peningkatan pada proyek berdasarkan penggunaan produksinya.

Dalam penggunaan praktis, SKT telah melihat sejumlah besar manfaat dari pengerahan OpenStack di Kubernetes termasuk:

- Instalasi Sederhana dan Mudah.
- Cluster Auto-Healing.
- Kemampuan untuk meningkatkan dan memperbarui OpenStack dengan dampak minimal terhadap layanan yang berjalan.
- Adopsi cepat dari metodologi rilis canggih, termasuk penyebaran biru-hijau, rilis canary.
- Selesaikan pengelolaan otomatis dependensi Python melalui isolasi kontainer.
- Pengelolaan rahasia dan konfigurasi yang aman.
- Roll-out pembaruan klaster yang cepat dan fleksibel.

SKT masih menguji pendekatan tersebut, tetapi secara aktif bergerak menuju penggunaan OpenStack-Helm mereka dalam produksi. Pada akhir tahun ini, SKT akan memiliki setidaknya tiga kelompok produksi, dengan yang keempat dan terbesar datang online pada tahun 2019. Kasus penggunaan ini termasuk:

- Platform Big Data (direncanakan untuk ditayangkan Q4 2018)
- Platform infrastruktur desktop virtual (produksi siap pada Q4 2018)
- Internal Private Cloud tujuan umum (direncanakan untuk ditayangkan Q3 2018)
- Infrastruktur jaringan telekomunikasi yang dibangun di atas fungsi jaringan virtual (direncanakan akan dibuka sekitar tahun 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.

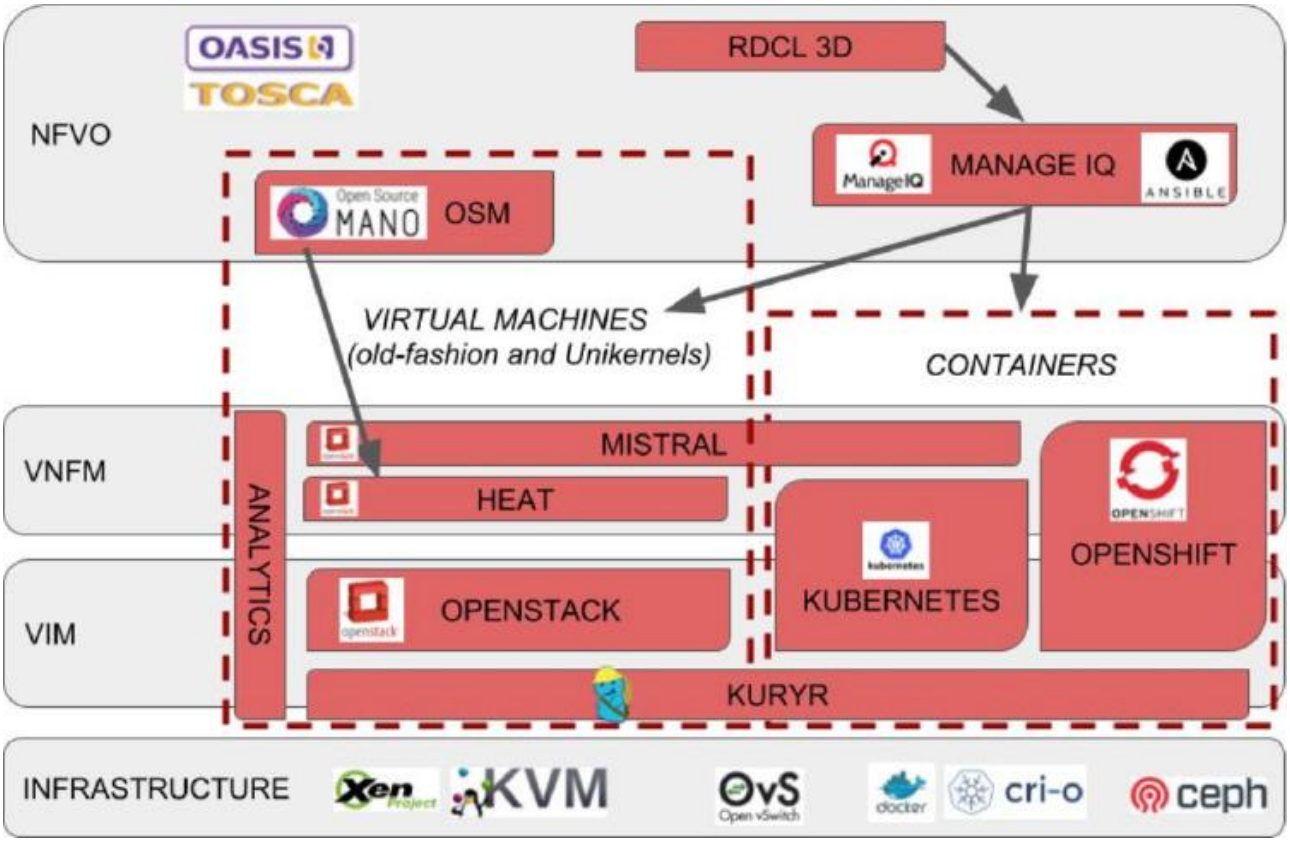
Secara keseluruhan, SKT menemukan bahwa Kubernetes membantu menyelesaikan banyak kerumitan penggelaran dan pengoperasian OpenStack. Menyederhanakan OpenStack memberi mereka pendekatan yang kuat untuk menghadirkan inovasi infrastruktur canggih untuk era 5G. Memfokuskan upaya pada Openstack di Kubernetes secara dramatis meningkatkan kemampuan internal mereka untuk menghadapi perubahan yang terus berubah menuju microservices dalam container dan menjadi infrastruktur penting untuk memberikan Artificial Intelligence, Internet of Things, dan Machine Learning.

Superfluidity

The Superfluidity projectterdiri dari 18 mitra dari 12 negara Eropa. Ini bertujuan untuk meningkatkan kemampuan instantiate layanan on-the-fly, menjalankannya di mana saja dalam jaringan (inti, agregasi, edge) dan memindahkannya secara transparan ke lokasi yang berbeda. SUPERFLUIDITY adalah proyek European Research (Horizon 2020) mencoba membangun blok infrastruktur dasar untuk jaringan 5G dengan memanfaatkan dan memperluas proyek open source terkenal. SUPERFLUIDITY akan memberikan konsep 5G berbasis cloud konvergensi yang akan memungkinkan kasus penggunaan inovatif di mobile edge, memberdayakan model bisnis baru, dan mengurangi biaya investasi dan operasional.

Untuk mencapai tujuan ini, konsorsium proyek bergeser dari legacy, aplikasi berbasis VM ke aplikasi Cloud Native containerized. Kuryr berfungsi sebagai jembatan antara mesin virtual OpenStack, dan Kubernetes dan layanan containerized OpenShift.

Proyek ini memanfaatkan ManageIQ sebagai pusat networks function virtualization orchestrator (NFVO), Ansible untuk penyebaran Aplikasi dan manajemen siklus hidup, layanan OpenStack termasuk Heat, Neutron, dan Octavia, dan Kubernetes melalui OpenShift untuk integrasi VM dan kontainer.



Dengan memanfaatkan playbook Ansible yang dijalankan dari alat ManageIQ, SUPERFLUIDITY menawarkan cara umum untuk menyebarkan aplikasi. Aplikasi ini pada gilirannya menggunakan fungsi orkestrasi awan yang disediakan oleh OpenStack Heat templates dan OpenShift templates.

Konsorsium itu menyebarkan cloud radio access networks (CRAN) 5G dan komponen mobile edge computing (MEC) dalam kontainer. Ini juga menyebarkan aplikasi throughput yang tinggi seperti streaming video di atas infrastruktur terdistribusi.

Berpindah ke pendekatan cloud native untuk pengiriman aplikasi memungkinkan pemasangan SUPERFLUIDITY yang cepat dan tangguh. Ini memungkinkan transisi yang mulus dari aplikasi dan komponen berbasis-VM ke kontainer, sementara tetap mempertahankan keserbagunaan untuk mengaktifkan VM untuk beberapa aplikasi tertentu. Contoh dari aplikasi ini adalah perlindungan keamanan khusus atau percepatan jaringan yang diperlukan oleh single-route input/output virtualization (SRIOV).

Dalam pengujian kinerja skala, SUPERFLUIDITY mampu meluncurkan sekitar 1000 pods dengan laju 22 pod / detik (dengan waktu yang diukur dari pembuatan hingga berjalan). Kinerja luar biasa ini dicapai dengan menjalankan OpenShift pada VM yang dikelola oleh OpenStack, dengan Kuryr bertindak sebagai driver jaringan pod untuk menghindari hit kinerja enkapsulasi ganda.

Select Another Case Study

IV. Kesimpulan

Selama beberapa tahun terakhir, karena kontainer telah menjadi alat penting bagi pengembang dan organisasi, OpenStack telah memanfaatkan desain modularnya dan komunitas luas untuk mengintegrasikan teknologi kontainer di berbagai tingkatan. Hal ini dapat dilihat baik oleh berbagai organisasi yang membawa kontainer maupun OpenStack ke dalam produksi, dan jumlah proyek yang bekerja bersama kontainer untuk memberikan kemampuan baru. OpenStack Foundation berkomitmen untuk memastikan bahwa teknologi yang muncul dapat dimasukkan dan digunakan dalam OpenStack, dan kontainer adalah contoh penting dari komitmen tersebut.

Untuk mempelajari lebih lanjut, kunjungi Containers Landing Page , di mana Anda dapat menemukan salinan dokumen ini serta tautan ke lusinan video yang berfokus pada integrasi OpenStack dan kontainer. Kubernetes SIG-OpenStackmemiliki saluran Slack, milis, dan pertemuan mingguan jika Anda terlibat langsung dengan komunitas yang membangun integrasi Kubernetes dan OpenStack.

V. Indeks Proyek Open Source

Airship

Airship adalah kumpulan alat open source yang dapat dioperasikan dan digabungkan secara longgar yang menyediakan provisioning dan manajemen cloud otomatis dengan cara deklaratif, berbasis di sekitar Kubernetes sebagai platform aplikasi.

Ansible

Ansible adalah alat orkestrasi yang biasa digunakan untuk mengerahkan dan mengelola instalasi OpenStack.

Cinder

OpenStack Cinder menawarkan penyimpanan blok sebagai layanan, menyediakan satu API yang didukung oleh lebih dari tujuh puluh driver penyimpanan yang mungkin berbeda.

Cloud Provider OpenStack

Cloud Provider OpenStack adalah implementasi antarmuka Kubernetes Cloud Provide. Ini memungkinkan kluster Kubernetes yang dikelola OpenStack untuk mengakses langsung penyimpanan dan sumber load balancer di awan OpenStack.

Calico

Calico adalah overlay jaringan dengan driver untuk Kubernetes dan OpenStack yang menampilkan L3-only routing.

Cyborg

Cyborg adalah proyek OpenStack yang menyediakan kerangka manajemen umum untuk akselerator perangkat keras termasuk FPGA, GPU, ASIC, dan lain-lain. Pekerjaan sedang berlangsung untuk memunculkan antarmuka perangkat keras umum ke pod.

Docker

Docker adalah kerangka kerja virtualisasi wadah sumber terbuka, yang digunakan untuk menjamu aplikasi kemas.

Helm

Helm adalah manajer paket resmi untuk Kubernetes. Penerapan aplikasi dijelaskan oleh Helm-Charts, yang dapat secara otomatis digunakan dan dikelola pada kluster Kubernetes.

Ironic

Ironis adalah layanan bare-metal OpenStack. Berjalan baik sebagai layanan mandiri atau sebagai sopir ke OpenStack Nova, dapat mengelola siklus lengkap sistem bare-metal, termasuk pendaftaran, penyediaan, pemeliharaan, dan dekomisioning.

Loci

LOCI adalah proyek OpenStack untuk membangun kontainer yang ringan dan sesuai OCI untuk proyek OpenStack.

LXC

LXC adalah antarmuka virtualisasi kontainer tingkat rendah yang memanfaatkan isolasi kernel namespace Linux dan teknologi lain untuk membuat runtime linux yang terisolasi.

Kata Containers

Kata Containers adalah implementasi standar dari Virtual Machines ringan (VMs) yang merasakan dan melakukan seperti kontainer, tetapi menyediakan isolasi beban kerja dan keuntungan keamanan dari VMs.

Keystone

Keystone adalah layanan OpenStack Identity yang menyediakan sarana untuk mengautentikasi dan mengelola akun pengguna dan informasi peran terutama untuk lingkungan cloud OpenStack, tetapi juga sebagai plugin untuk lingkungan lain, termasuk Kubernetes.

Kolla (Containers)

Kolla (Containers) adalah proyek OpenStack untuk membangun kontainer untuk setiap layanan OpenStack. Ini termasuk sistem build dan templating yang canggih, dan mampu membangun kontainer dari sumber dan paket pada berbagai sistem operasi host.

Kolla Ansible

Kolla Ansible adalah proyek OpenStack yang menggunakan Ansible untuk menyebarkan dan memelihara instalasi OpenStack penuh menggunakan kontainer Kolla.

Kubernetes

Kubernetes adalah sistem orkestrasi kontainer yang memberikan aplikasi yang kuat dan sangat tersedia di atas infrastruktur cloud.

Kuryr

Kuryr adalah proyek OpenStack yang menyediakan overlay jaringan Neutron ke runtimes kontainer, termasuk Docker dan Kubernetes. Ini bertujuan untuk menjadi "integration bridge" untuk jaringan kontainer dan VM.

Magnum

Magnum adalah proyek OpenStack yang menawarkan platform kontainer yang dikelola sebagai layanan, termasuk Kubernetes, Docker Swarm, Mesos, dan platform DC/OS. Hal ini mampu menciptakan platform aplikasi penyewa yang terisolasi melalui user-facing API sederhana.

Neutron

Neutron adalah layanan jaringan software-defined OpenStack, menawarkan satu API untuk memberikan infrastruktur jaringan dinamis yang didukung oleh puluhan driver jaringan.

OpenStack Ansible

OpenStack Ansible adalah proyek untuk membangun layanan OpenStack ke dalam kontainer LXC, dan untuk menyebarkan dan mengelola instalasi OpenStack di dalam layanan containerized tersebut.

OpenStack Helm

OpenStack Helm adalah proyek OpenStack yang menyebarkan dan mengelola siklus hidup OpenStack dan infrastruktur pendukung di atas Kubernetes (misalnya Ceph dan MariaDB), menyediakan produksi siap digunakan, untuk berbagai kasus penggunaan dari pengerahan small edge ke kantor pusat besar. Memanfaatkan sistem manajemen paket Helm. OpenStack Helm memiliki dukungan untuk manajemen beban kerja baremetal (Ironic) dan virtual (Nova/KVM), dan merupakan image agnostik yang mendukung kedua kontainer LOCI dan Kolla.

Qinling

Qinling adalah proyek OpenStack untuk memberikan Fungsi sebagai Layanan. Qinling mendukung platform orkestrasi kontainer yang berbeda, seperti Kubernetes dan Docker Swarm, serta backend penyimpanan paket fungsi yang berbeda seperti file-store lokal, OpenStack Swift, dan S3.

Triple-O

TripleO adalah proyek yang bertujuan untuk memasang, meningkatkan, dan mengoperasikan cloud OpenStack menggunakan layanan cloud OpenStack sebagai fondasi - membangun Nova, Ironic, Neutron, Heat, dan Ansible untuk mengotomatisasi manajemen cloud.

Zun

Zun adalah layanan Kontainer OpenStack. Ini bertujuan untuk menyediakan layanan API untuk menjalankan kontainer aplikasi tanpa perlu mengelola server atau kluster.

VI. Penulis

Anggota dari 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 Systems GmbH
- Sangho Shin, SK Telecom
- Davanum Srinivas, Huawei
- Luis Tomás, Red Hat, Inc.
- Sam Yaple, Verizon Digital Media Services
- Mikhail Fedosin, Red Hat, Inc.
- Flavio Percoco, Red Hat, Inc.

Editor

- Brian E Whitaker, Zettabyte Content LLC

Translators

- suhartono, <cloudsuhartono@gmail.com>