Sponsored Report

Container Networking for Enterprises: From Trial to Production

nuagenetworks
From Nokia

sdx central®

The Trusted News and Resource Site for SDx, SDN, NFV, Cloud and Virtualization Infrastructure

Table of Contents

Connect anything, anywhere.

Nuage Networks Virtualized Services Platform (VSP) uniquely provides secure cloud and network automation across all major application platforms and offers you uncompromised networking at massive scale for all your workloads and all your locations.

- Virtual Machines
- Containers
- Bare Metal
- Branch
- Cloud

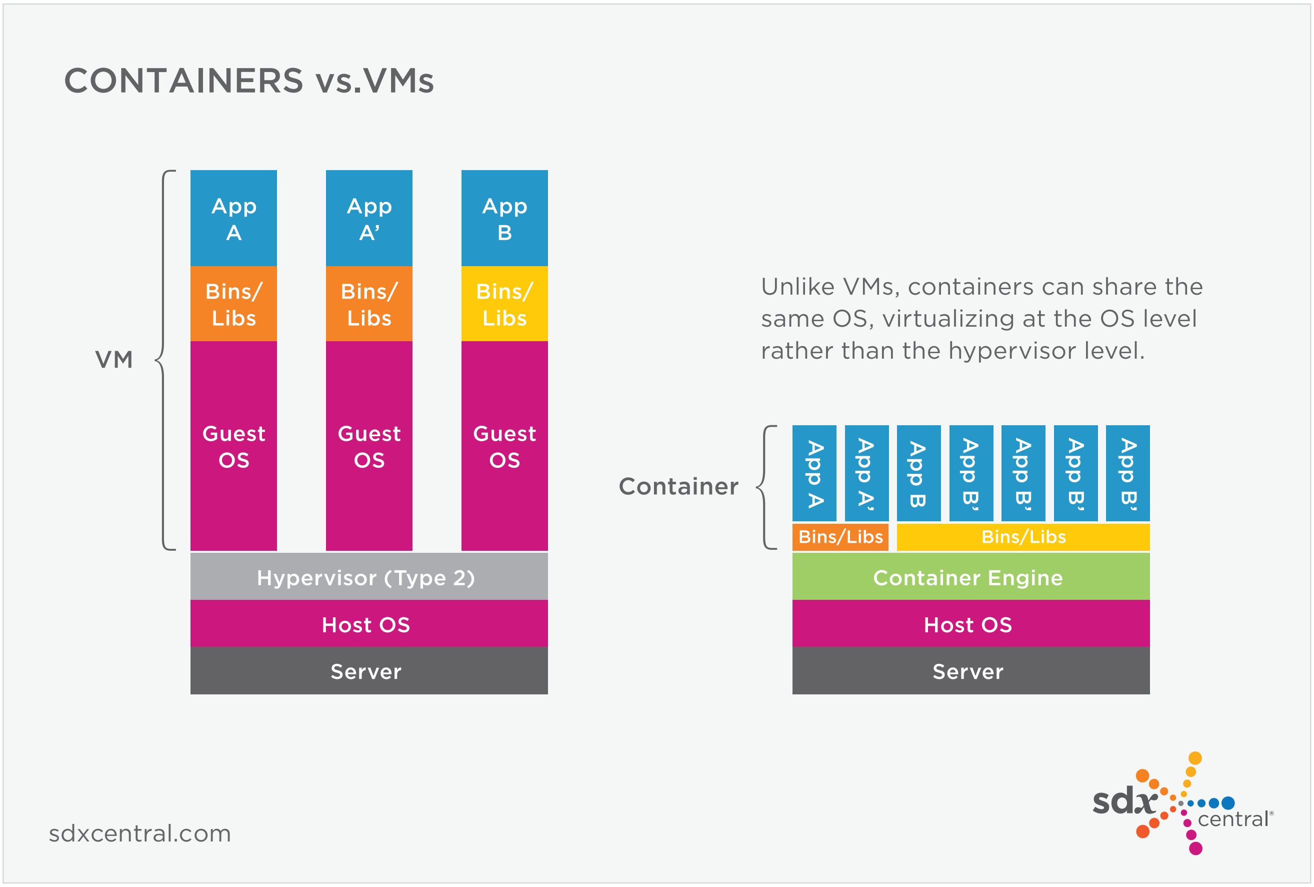
The only virtual networking solution your IT organization will ever need.

Seamless secure network automation.
Simplified IT. Accelerated application deployment.
www.nuagenetworks.net

nuagenetworks
From Nokia

Introduction: Containers Versus Containers at Scale

Developers love Linux containers because they represent a simple way for them to run their code anywhere. Docker Inc. has helped drive the trend by creating an easier way to use Linux container technologies. Containers can be deployed on top of a virtual machines or on bare metal servers. However, as much as developers enjoy that level of flexibility, some operational challenges come with that flexibility. In fact, even though containers can eliminate the need for virtual machines, most IT organizations opt to run them on top of virtual because they don't yet have the tools needed to manage them on bare metal servers.



Over time, we expect a lot more containers running on bare metal servers because that allows increased utilization of IT infrastructure. But for the foreseeable future, the one thing IT organizations can count on is the simple fact that containers will be running on all different platforms, both on-premises and in the cloud.

As this occurs, there will naturally be a pressing need to not only interconnect those containers, but also integrate container applications with legacy applications residing on existing networks. The challenge facing IT organizations now is deciding what's the most efficient approach to achieving that goal.

Most IT organizations are not prepared to manage containers that will be strewn across the enterprise. Before they realize it, thousands of containers end up running on top of virtual machines, bare metal servers and new lighter weight container-as-a-service (CaaS) environments. Each of those container deployment models will need to share access to a common secure network capable of scaling to meet those requirements.

While sharing access to a common secure network may seem straight forward enough, that plan can turn out to be anything but when deploying containers at scale. Microservices enabled by containers can share resources on the same cluster rather simply. But once microservices begin to span multiple clusters, competition for networking resources starts to escalate. To make matters even more interesting, containers show up in two classes of applications. Stateful applications that require access to a database tend to have longer running containers that are only replaced when new functionality is added. Stateless applications based on containers are much more ephemeral. According to New Relic, a provider of monitoring tools, the average Docker container in those application scenarios has an average lifespan of less than 10 days. But those numbers simply reflect the fact that there are, for now, a lot more stateless applications making use of containers than stateful ones. The whole point of embracing microservices based on containers is to inject resiliency and flexibility into the IT environment. That goal won't be achieved if the networking environment doesn't scale.

IT organizations would also be well advised to remember that containers employ different OS libraries running on different platforms. A container running on a Linux server can't be ported to an instance of Windows Server 2016 or Microsoft Azure. The only way those container applications will interact with one another is via the network.

Finally, container networking needs to address the ongoing monitoring of those containers in addition to load balancing across multiple hosts. Many IT organizations are about to discover their existing monitoring tools can't see containers and that the load balancing software they rely on needs to be either upgraded or replaced.

A Cloud-Enabling Technologies Market Monitor published by 451 Research forecasts that the application container market will grow to $2.7 billion by 2020 from $762 million in 2016. That represents a relatively small percentage of the overall IT market. But a 40 percent compound annual growth rate will make containers a force to be reckoned with. Put it all together, and it's clear that successfully making the transition to container networking will require a lot of forethought.

Types of Container Networking

There are multiple types of container networking ranging from the simple to complex:

- **Links:** These are hard-coded network links between containers. Rarely used anymore, they were the first form of basic container networking.
- **Container-Mapped Networking:** One step of sophistication above a hard-coded link, network maps allows containers to make use of namespaces to access another container.
- **Internal Networking:** In these instances, a container doesn't have a dedicated network interface, but does have a network stack that can be employed mainly for application testing purposes.
- **Bridging:** Makes use of Linux bridging to allow containers residing on the same host to communicate.
- **Host Networking:** In this model, every new container is automatically assigned a namespace that allows the container to communicate via a network stack residing on the host system.
- **Overlays:** This approach creates a tunnel through which containers residing on different hosts can communicate with one another directly.
- **Underlays:** Containers can also be configured to access physical underlays directly using, for example, a Media Access Control Virtual Local Area Network (MACvlan) or an IP protocol.

State of Container Networking

Clusters running containers today make use of various methods for interconnecting containers operating on the same cluster. But many of the technologies being employed to network those containers together are still relatively immature. Because of that issue, IT organizations that are deploying containers in a production environment typically opt to either deploy those containers on top of a virtual machine that has access to a software-defined virtual network overlay or make use of a platform-as-service (PaaS) environment that provides a layer of abstraction above any number of networking options. There are several options, both commercial and open source, including Flannel from CoreOS, Nuage VSP from Nuage Networks (division of Nokia), OpenShift SDN from RedHat, Calico from Tigera (spin off of Metaswitch), libnetwork from Docker, WeaveNet from WeaveWorks, OVN from VMware, Contiv from Cisco, and OpenContrail from Juniper.

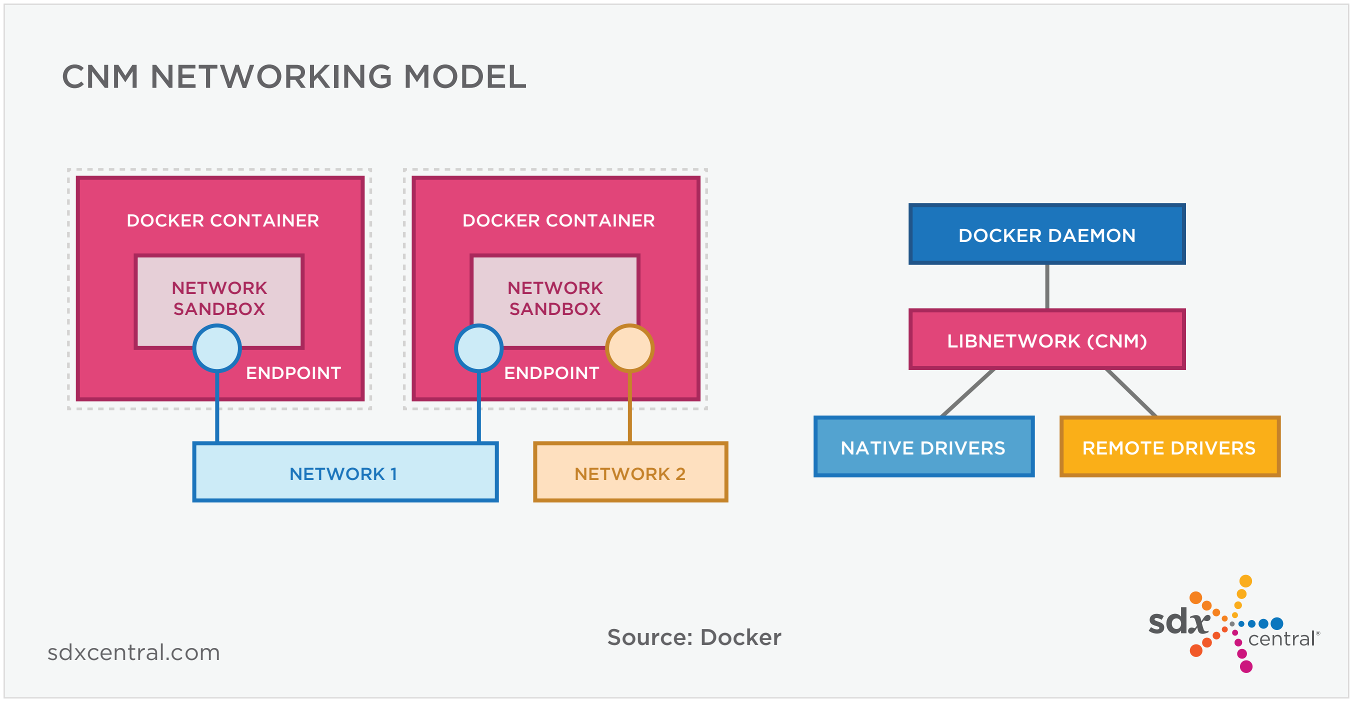
	Product	**Vendor**
flannel	Flannel	CoreOS
nuagenetworks From Nokia	Nuage Networks VSP	Nuage Networks from Nokia
OPENSHIFT	OpenShift SDN	RedHat
	Calico	Tigera
docker	Libnetwork	Docker
	WeaveNet	WeaveWorks
	OVN	VMware
Contiv	Contiv	Cisco
OPENCONTRAIL	OpenContrail	Juniper

Container Network Model (CNM)

Fundamentally, network containers invoke two different interfaces today. The first is the Container Network Model (CNM) developed by Docker. CNM most often manifests itself in libnetwork (seeded by Docker's acquisition of SocketPlane), a framework employed by Docker to provide an interface between the Docker daemon and network drivers.

Written in the Go programming language, libnetwork is an amalgamation of the networking code from both libcontainer and Docker Engine that Docker has been expanding to create a multi-platform library for networking containers. libnetwork employs a driver / plugin model that makes use of CNM to abstract away the underlying complexity of implementing all the drivers. A network controller pairs a driver to a network and each driver, but multiple drivers can be used concurrently with containers connected to multiple networks.

CNM enjoys support from a variety of open source projects as well as vendors such as Nuage Networks[1], Cisco and VMware. There are, however, efforts underway to extend container networking functions beyond the basic capabilities provided in libnetwork plugins. Docker, for example, is working through an implementation of Docker Networking that relies on a network overlay to connect containers across clusters.



Container Network Interface (CNI)

The second major container networking interface is the Container Network Interface (CNI) originally developed by CoreOS. The three primary components of CNI consist of a specification that defines an API between runtimes and network plugins to set up a container network setup, a series of plug-ins that address various use cases, and a library of code written in the Go programming language that developers can embed in their applications. CNCF expects vendors and IT organizations to develop plug-ins by making use of schema defined in JSON.

The Cloud Native Computing Foundation (CNCF), an arm of The Linux Foundation, has agreed to make CNI a project and will spearhead it to advance and standardize container networking, just like they did with Kubernetes as an orchestration framework. The primary technical objection against CNM is the fact that it is optimized for the Docker container runtime while CNI is more open. Kubernetes advocates contend that Docker wasn't ready to accommodate changes that would be needed to make CNM more runtime independent.

Specifically, Kubernetes advocates note CNI doesn't require daemons and that it's simpler to wrap a CNI plugin using a shell script. In addition, these advocates contend Docker drivers are difficult to map to other control planes. Specifically, drivers are not told the name of the network to which a container is being attached, making it difficult for a driver to map back to any concept of network that exists in another system.

[1]Available from Github: https://github.com/nuagenetworks/nuage-libnetwork

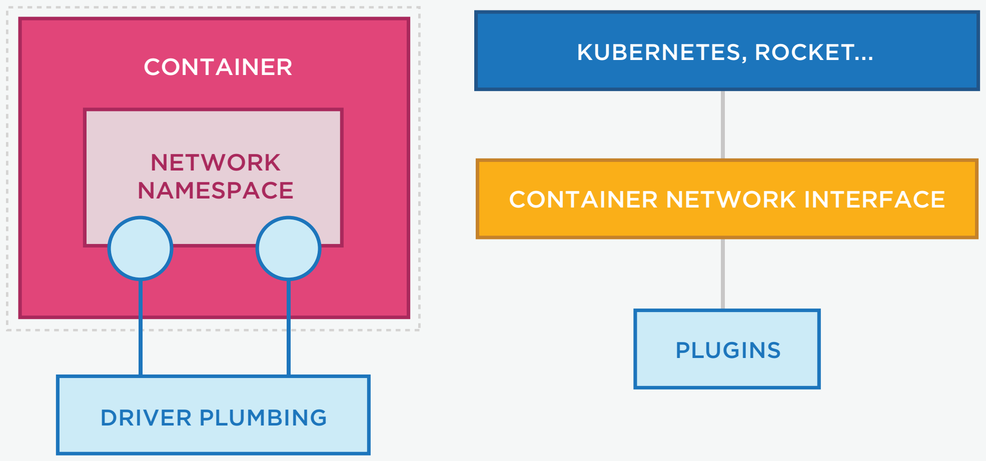
CNI also enjoys support from Mesospshere, which has developed a Data Center Operating System that employs open source Mesos cluster running Marathon container orchestration. In that model, Mesosphere has created a plugin that adds a software-defined networking overlay. VMware also is supporting CNI in beta form through a version of its NSX-T (multi-hypervisor overlay networking solution). In addition, Nuage has open sourced it's versions of CNI plugins for both Mesos and Kubernetes[2].

In all cases, these clusters will expose application programming interfaces (APIs) that will simplify support for container clusters for providers of networking solutions. It's not clear yet whether Docker and CNCF will ever see eye to eye on network interfaces. But under the leadership of newly appointed Docker CEO Steve Singh, the company will likely warm up to the idea of allowing customers swap out elements of the container stack as they see fit, given Singh's recent signaling to the market of opening up the Docker ecosystem. Beyond allowing third-party plugins within libnetwork, the container ecosystem is watching to see if Docker will move to CNI, or whether CNI and CNM come together in some form.

Both CNM and CNI create some interesting options when it comes to container networking. Given the relative immaturity of container networking technologies, IT organizations are understandably hesitant to standardize on a specific vendor offering. These interfaces should reduce the pain associated with switching container networking technologies as well as make it simpler for providers of network overlays that span multiple computing models to create plug-ins for container clusters.

Longer term, however, enterprise IT organizations and service providers are likely to bring a significant amount of pressure to rationalize the two interfaces that provide essentially similar value. In fact, as IT organizations move to standardize on one interface or another, many networking vendors will have a vested interest in both reducing the cost of supporting two separate interfaces.

CNI NETWORKING MODEL



sdxcentral.com

Source: Docker

[2]Available from Github at https://github.com/nuagenetworks/nuage-cni

More users. More applications. More endpoints. More complexity.

Make your apps and services cloud-ready with the first complete software-defined security solution for multi-site cloud networks.

Nuage Networks Virtualized Security Services (VSS) lets you use SDN to automate security policies across cloud, datacenter and branch networks. It helps you build a cloud environment that can handle fast change and increasingly sophisticated attacks.

- **PREVENT** security incidents with micro segmentation for all workload types
- **DETECT** threats and compliance issues with contextual network visibility and security analytics
- **RESPOND** to incidents with automated remediation

Don't let security and compliance issues hold you back. Choose Nuage Networks VSS and embrace the cloud with confidence.

Software-defined security
for your datacenter and beyond
Learn more at nuagenetworks.net/VSS

nuagenetworks
From Nokia

Containers and Their Role in Larger DC Ecosystem

When it comes to building a network overlay, a myriad of options exist. In basic terms, a layer of abstraction that creates a network tunnel between endpoints over a virtual network provides an efficient way for those endpoints to communicate across a physical network. As adoption of microservices enabled by containers continues to expand, the number of endpoints distributed across an extended network will significantly increase.

Too many organizations underestimate the complexity associated developing and running their first container network. One of the simplest ways to begin is to employ Swam Mode when setting up multiple Docker hosts. Assuming each cluster has access to a high-speed network underlay, it should be relatively straightforward to assign a port to each application address running on those hosts. Similar methods exist within Mesos and Kubernetes. While getting networking going is easy, in a production environment, many developers and IT operations teams fail to account for the latency-sensitive characteristics of many microservices.

In many cases, microservices will employ routing techniques to communicate with one another. But just as often microservices will also need to communicate with legacy applications that could reside locally or half-way around the world. Network overlays combined with automation and SDN can help provide a mechanism for microservices to interact with legacy applications without adding further to the IT workload in reconfiguring existing physical network. Regardless of the approach used, an upgrade to the physical network underlay may be required to successfully deploy containers in a networking environment. This is because many IT operations teams often underestimate just how many microservices they must deploy on their networks, and how latency-sensitive some of the services might be. Because modern applications built using microservices consist of many more discrete elements, an application could consume 10s, 100s or more microservices, with each service collaborating with their peers and requiring multiple round-trips before the results of an API are returned.

Today, the average physical server might host 20 or more virtual machines. When containers are deployed on top of those virtual machines, they may add a dozen or more microservices trying to access a virtual switch. on those virtual machines. But as more containers are deployed directly in bare metal servers, hundreds or thousands of containers will likely ride on a physical server—all trying to share the same 10G Ethernet card connection to the physical underlay. Over time, data centers will see hundreds of thousands (or more) of containers running on a mix of bare metal and VM servers, with hundreds or thousands of distinct microservices, with varying SLA requirements and networking needs. Managing that level of interaction at scale will tax both the patience of network administrators as well as the physical underlay. Network administrators need to find ways to optimize flows between containers, microservices and legacy applications using policy-based engines embedded in the network overlay.

In fact, many IT organizations will be forced by the rise of those containers to embrace advanced forms of network management that rely on machine learning algorithms to optimize traffic flows across an extended network. It won't be feasible for network administrators to manage that level of complexity on their own.

In the meantime, many service providers are leading the charge when it comes to deploying containers alongside Network Function Virtualization (NFV) software. NFVs today assume there is some form of a virtual machine in place to run networking software. In the future, however, networking software could just as easily be packaged in a series of containers. No one approach will supplant the other. Rather, service providers will likely deploy a medley of containers and NFVs depending on the requirements of any given service. Containers will be viewed as a lighter weight instance of NFV software over time. In those deployment scenarios, carriers such as AT&T have already made it clear they plan to deploy a common management plane across NFVs based on virtual machines and containers. In most enterprise scenarios, internal IT organizations will naturally look to IT vendors to provide similar management capabilities using a combination of open source software they have curated and proprietary software through which they add value.

Role of Open Source in Container Networking

The existence of container networking interfaces is also one of the driving forces behind a variety of promising open source projects. One particularly popular example is the Project Calico effort that provides a control plane for building a software-defined network spanning container clusters and existing IT environments.

Using a fork of the BIRD routing protocol stack (though recent releases also added experimental support for GoBGP as an alternative), Project Calico extended basic routing capability by adding in the consensus algorithm as well as the Network Policy application programming interface (API) that Google developed for Kubernetes. This approach eliminates the need for a network overlay altogether. Instead, Calico routes packets from the workload onto the underlying IP network without any extra headers. The only exception would be when crossing availability zone boundaries in the public cloud. In that circumstance, Project Calico can make use of lightweight encapsulation, including IP-in-IP and VxLAN, across either an IPv4 or IPv6 network. In addition to support for CNI and CNM, Calico comes with a variety of plug-ins for OpenStack distributions as well as cloud orchestration engines such as Kubernetes, Mesos, and Docker.

The Project Calico approach enables IT organizations to microsegment networks in a way that makes it possible to deploy fine-grained firewalls (based on Linux iptables) to better secure microservices at a level of scale that no other networking approach can hope to match. Project Calico is now managed and powered by Tigera, a start-up that was spun out of Metaswitch to focus on container networking.

One of the challenges networking professionals face when it comes to container networking is that there are multiple open source projects to track. The CNCF, for example, has started a Linkerd proxy project for containers that encompasses service mesh functionality. Meanwhile, Google and IBM in collaboration with Lyft have launched the Istio project that combines technologies all three companies have developed independently to create similar service mesh for containers. Today, however, Istio only supports Kubernetes clusters. Regardless, both of these service mesh solutions provide higher-level services such as load-balancing, distributed tracing, service-aware routing, encryption and both rely on an underlying container networking framework

No standards body has been appointed yet to reconcile all these efforts. There's no doubt that in time much of the functionality being made available via all these projects will be incorporated into a variety of network virtualization and SDN offerings. And we've already seen numerous SDN vendors open-sourcing their network virtualization plugins to help popularize their solution. Nevertheless, most IT organizations would be well advised to not deploy raw open source bits in a production environment given many of them are immature, at least not without significant upfront testing, a source of support from the vendor or a credible third-party who can support the code.

Requirements for Production-Grade Container Networking

When it comes to the future of networking in the enterprise, IT organizations need to consider issues that go well beyond what may be convenient for any individual developer to implement. Those issues include:

Scale: Organizations need to figure out a strategy that can adapt well to a high-volume of endpoints with rapid container activations (and deactivations). In addition to being extensible, modern networks need to be able to scale to a point that not only meets today's requirements, but also future ones.

Hybrid Environments: It's not practical for most organizations to implement and manage multiple networking platforms or networking strategies in a production environment, picking one flavor for bare metal, another for VMs and yet another for containers. Modern networks need to be able to support a broad range of legacy applications and virtual machines in addition to containers and whatever else may come down the proverbial pike next.

Security: IT organizations need to implement consistent security policies across a diverse range of applications running across networks that are microsegmented in a way that prevents malware from infecting the entire enterprise. That means detecting potential security threats and violations of compliance mandates; it also means providing the tools needed to visualize and inspect ongoing flows, as well as easily implement access controls and automate the remediation process in near real time.

Manageability: IT organizations need a modern networking framework that glues together infrastructure, orchestration platforms, applications and legacy assets in a way that is simple to manage. IT organizations can no longer afford to hire and train specialists for separate application silos.

The single biggest container networking issue that IT leaders should resolve sooner than later has nothing to do with the bits themselves. Developers are increasingly taking advantage of APIs to programmatically invoke IT infrastructure. This is already routine in public clouds that expose compute, storage, and networking resources via an API. Developers are going to expect similar functionality to be placed at their disposal on premise. IT leaders are going to need to find a way to provide self-service networking capabilities to developers within the confines of policies set up by the networking team. In effect, IT organizations will, in many cases, need to rethink delineation of duties across their IT environments. Failure to address this issue will simply encourage developers to do an end run around the internal IT organization by deploying more applications in a public cloud regardless of whether it makes economic sense or violates any number of enterprise security and compliance policies.

In fact, the rise of containers is going to force many network administrators to come to terms with DevOps. In the age of digital business, there's more pressure than ever on IT organizations to be more agile. It only takes a few minutes to spin up a virtual machine. But it can still take weeks to provision the associated networking resources. IT organizations and service providers are investing massive sums of money in multiple forms of network virtualization to make networks more responsive to changing business requirements. Once those virtual networks are rolled out, the next logical question becomes: To what degree can a network administrator enable self-service provisioning on those networks? In the case of end users, that may mean creating a portal through which they can select a class of networking services. For developers working with microservices and containers, the expectation is that the network itself is a programmable entity that developers can directly invoke via an API.

Developers will also expect to access levels of visibility that reveal any network segment that might be adversely impacting the performance of any given container and associated microservices. In fact, many network administrators may not be prepared for how latency-sensitive many of those microservices applications will be. Modern developers are going to routinely expect networks to support millisecond response times across a wide range of transactional and customer engagement applications. Network administrators should expect to spend a substantial amount of time in war room conversations with developers trying to determine where exactly a specific application bottleneck might lie. As is often the case with developers, there's a tendency to blame the IT operations team until proven otherwise. Even when it might become apparent that the issue is the responsibility of the IT operations team, the team must figure out whether the problem is a network issue or one involving the way a server or storage system connected to the network is configured.

One thing that is certain is that the provisioning of network resources is about to become part of a much larger DevOps conversation. This discussion will include everything from how APIs are published and maintained to how the IT organization itself is organized. Instead of silos of networking, storage, and compute teams within the data center, many IT organizations are embracing organizational models that align cross-functional teams of developers and IT operations staff around specific classes of applications.

Those cross-functional teams will not only need to reconcile performance issues, but security and compliance concerns as well. Keeping track of containers that have only been running for a few hours creates a substantial challenge, and most auditors today are not going to care that containers are more ephemeral than virtual machines. IT operations teams will still be accountable for documenting those interactions. Because of that issue, policy-based approaches to networking are critical because they demonstrate to auditors that there's a consistent approach to managing IT that can be documented. Unless an IT organization can present that documentation, the auditor more than likely will spend weeks trying to create that documentation. The real goal, of course, should be to enable auditors to get in and get out of an auditing engagement as quickly as possible.

With these enterprise-level requirements for production-level containers, it's imperative that CIOs looking to put in place their high-level IT strategy for containers understand all facets of their decision and evaluate the choice. They should consider a mix between commercial solutions and open-source solutions for each element in their container infrastructure, from the run-time OS to the networking foundations.

Container Networking Skills

One upside of the rise of container networking is that demand for networking professionals with expertise in this area is about to substantially increase. There's already a shortage of IT personnel with the advanced skills needed to drive modern applications. IT leaders should expect to see organizations such as CNCF alongside vendors pouring resources into container training and related certifications.

But as any IT professional knows, there's no substitute for hands-on experience. In terms of being able to command a premium for their services, many networking professionals would be well advised to download various forms of container technologies to expand their skill sets. After all, whenever a new technology starts to receive mainstream attention, it's not too long before demand for IT professionals who have not kept their skills current begins to wane. Network administrators who have programming skills, for example, already command a salary premium over those who still rely on command-line interfaces to manually interact with one networking resource at a time. As networking evolves, it has become apparent that the sheer volume of networking services that must be provisioned and updated will soon make legacy approaches to managing networks nothing less than archaic.

Ideally, more enlightened organizations make available the time IT professionals will need to come up to speed on containers in general and container networking specifically. But just in case, there's no shortage of industry conferences and tutorials describing the inner workings of containers as well as online demos and webinars describing in detail how to create a container network.

Rather than waiting for the organization they work for to make that training available, history has shown time and again that IT professionals who invest in themselves make more money over time than those who don't update their skills. After all, networking professionals who keep their own skills current are masters of their own destiny.

Summary

IT leaders would be well advised to remember that while usage of microservices based on containers are growing like wildfire, they represent only one computing model across a continuum. Not only are legacy applications and virtual machines likely to be parts of the networking landscape for years to come, additional computing models are on the horizon. Serverless-computing models, for example, make use of event-driven architectures to programmatically make compute, storage and networking resources available on demand. Ideal for short running applications, stateless applications using containers will soon migrate to serverless computing frameworks. Of course, that should still leave thousands of stateful applications running on containers. Add in new types of bare metal servers based on graphical processors units (GPUs) and field programmable gate arrays (FPGAs), and the number of places where containers can run will exponentially expand.

In fact, a strategic approach to networking that makes it possible to invoke all these services simultaneously will be the key to combining multiple clouds in a way that drives hybrid cloud computing initiatives. The real goal going forward is not to move every application to one specific type of computing model. Enterprise IT is already operating across multiple clouds. The future of enterprise IT will be based on a federated model that relies on SDNs to provide access to applications wherever they reside. IT organizations will be able to employ every type of computing model based on what makes the most economic sense at the time rather than hosting applications based on IT philosophy. Just as importantly, as applications evolve and change over time, IT organizations should put their confidence in management planes and networking architectures designed from the ground up to be extensible. IT organizations should also be wary of approaches that are hard wired to a particular type of physical underlay. These approaches not only lock them into a particular vendor, but also makes it more difficult to embrace additional computing models as they mature.

Container networking is not much different from traditional networking as there is continued reliance on routing underlays and various forms of network virtualization. IT leaders should take heart that even IT behemoths such as Amazon Web Services (AWS) are just now getting around to providing native networking services for containers on their cloud services.

What will be very different, however, is the level of networking scale microservices enabled by containers will drive. Most networking teams are hard pressed to keep pace with existing networking demands. The rise of microservices based on containers are about to test the skills on networking professionals in more ways than most of them would have thought possible as little as a year ago. In making the final decision on container networking, CIOs should ask whether the decision is around cloud networking and ensure that their cloud networking strategy accommodates containers, VMs, and bare metal servers both on premises and in the cloud while supporting their security strategy.

Embrace DevOps with SDN and policy-based automation

DevOps can greatly accelerate application development and delivery processes. It can also enable you to better align applications and IT services with business processes and customer activity.

Speed your transition to DevOps with Nuage Networks Virtualized Services Platform (VSP). Our SDN portfolio lets you use policy-based automation to quickly incorporate DevOps within your IT processes.

Nuage Networks VSP simplifies networking for virtualized applications and hybrid clouds. It gives you the freedom to integrate legacy applications, virtual machines and container-based microservices that work for your business.

Learn more:
nuagenetworks.net/containers

nuagenetworks
From Nokia

@nuagenetworks

SDNCentral, LLC
4677 Old Ironsides Dr, Ste 180
Santa Clara, CA 95054
www.sdxcentral.com

nuagenetworks
From Nokia

sdx central®

The Trusted News and Resource Site for SDx, SDN, NFV, Cloud and Virtualization Infrastructure

Rev A