A Capacity for Containerization

Before bringing containers into the data center, carefully consider how you'll deploy them and how they'll affect your capacity-management plans.
The Route to Container Success

Capacity planning gets more difficult when containers are put into action. Or does it get easier?

There’s a fair bit of mystery about just how containerization affects an IT infrastructure. This handbook examines the case from several angles, beginning with an article about how containers change an organization’s approach to capacity, storage and other key IT resources. SearchITOperations.com’s Meredith Courtemanche discusses these issues with several professionals, sharing their concerns about how containers are—and are not—changing capacity management.

Cloud consultant Tom Nolle, meanwhile, explores the important links between containerization and microservices. Being responsive, especially in the era of mobile workforces, means IT organizations need to be more tactical. Effective event-driven IT, Nolle says, simply isn’t possible without bringing container and microservices technologies into the infrastructure mix.

In addition, TechTarget’s Beth Pariseau writes about the two schools of thought on how best to deploy containers: on VMs or on bare metal? She talks with IT professionals in both camps, and explains the pros and cons of each approach.

As more businesses contemplate containers, decision makers will need to prepare. There will be important choices to make about using containers to achieve the goals of lower overhead, faster boots, simpler updates and more efficient use of infrastructure.

Phil Sweeney
Senior Managing Editor
Data Center and Virtualization Group
TechTarget
Containers and Their Capacity Questions

With respect to capacity planning, containers boast two advantages over virtual machines: lower overhead and free movement across infrastructure types.

Containers share hardware as well as the operating system. They use less storage and memory than a VM. Dozens of VMs fit on one average physical host, while the same host could run hundreds of containers. In actual implementations, however, most IT teams run containers on VMs rather than bare metal.

“[Containers] need little extra resources to run over the bare application,” said Gustavo Muslera, system administrator at LACNIC, the Regional Internet Registry for Latin American and Caribbean regions. But, he added, “you still need a VM to run them.”

Muslera sees simplicity, portability and app-orientated operations in today’s container ecosystem. He uses Docker containers for pre-production app evaluation and to test apps in a continuous integration scenario.

While containers aren’t right for all applications, they appeal to him as a Linux administrator. He likes their portability from development to test, staging and production; the streamlined backup; and the replication from a shared base image.

Containerization changes how Muslera approaches storage, network and compute capacity planning problems, even when he’s not actually using containers for a deployment. To him, the essence of the problem is now different.

When building physical infrastructure to host virtual servers, Muslera implements Docker-like concepts, including mounted storage volumes and forwarded ports, to design the architecture to accommodate future migration or scaling into containers. Muslera also endeavors to develop a fast path to move services away from the most heavily used servers,
Containers ease the concerns around updates and patching, concerns that grow as infrastructure gets more distributed.

The online payments company’s efforts to modernize include growing from three data centers in the U.S. to a global presence and re-architecting applications to take advantage of a horizontally scalable, repeatable model. Containers alleviate the concerns around updates and patching—concerns that only grow as infrastructure becomes more distributed, Dankar said.

Dankar’s team creates platform-as-a-service capabilities on the company’s OpenStack private cloud infrastructure for PayPal’s developers, who are its clients. The team deploys most applications on pools of VMs, and, to avoid disruption, containerizes apps. Those are mapped to the same or equivalent VMs. While it isn’t as efficient as bare metal, more than one container deploys and runs in that single VM. Sidebar containers—essential services for the application—share it.

As PayPal moves from a few hundred instances of containerized apps to eventually converting 10 million lines of code in various stacks, Dankar expects the lower-overhead containers to reduce the company’s spending on IT infrastructure. His focus for now is on operational concerns, such as how containers change the patch process and how they should be monitored.

IMPROVING WORKFLOW

Resource overhead isn’t a concern for every container adopter.

“Capacity wasn’t important. Stability was the primary driver,” said Stephen Eaton, infrastructure technical lead at Dealertrack Technologies, a holding of Atlanta-based Cox Enterprises.

Encapsulating applications in containers that float over infrastructure made the workflow...
easier for the entire IT group. However, as he ramps up containerization—the goal is 80% of the group’s apps on containers within a year—Eaton will be closely watching network-attached storage performance. With five times as many apps using the storage resources, will there be latency with logs or scaling that necessitates changes?

Jay Lyman, cloud management and containers research manager at 451 Research, bemoans the lack of best practices and standards in containerization. A maximum size for containers or strategies to prevent sprawl, he said, would be especially useful. These missing guidelines are among the reasons ops teams choose to put containers in VMs, he said.

Containers also change the equations for dynamic and static load balancing. While container-monitoring capabilities are not yet close to those available for virtualization, log-monitoring tools such as Sysdig and Splunk are working on the visibility issue with admin-friendly dashboards. Eaton uses Sysdig cloud to monitor each node in every app and says it’s always open on his screen.

These operational problems are just beginning to be sorted out, Lyman said. That means containers will run on VMs for at least the next two to three years, he predicts, until technological maturation and capacity utilization combine to pressure organizations to cut out the middle man. Administrators and DevOps engineers foresee a future where orchestration from Apache Mesos, Docker Swarm or Google Kubernetes replaces VM-based container management.

TALK TO YOUR VENDORS
While the new players—CoreOS, Docker, Kubernetes, Mesosphere and others—attract a lot of interest, big IT infrastructure vendors are responsive to containers.

“Most of the big vendors have a container strategy,” Lyman said, and your networking, storage, OS and management software vendors already know your organization’s infrastructure setup. “It can make change less arduous if you’re sticking with some existing stuff,” he said.

Muslera suggests evaluating the OS carefully, as older-generation kernel distributions
don’t always work well with Docker. Applications that need special kernel modules may be incompatible with containers for security reasons.

It’s not advisable to run containers on non-cloud infrastructure. “You won’t get the advantages of containers, and you might incur the penalties,” Lyman said. But he also noted that this is unlikely to stall containers, as organizations are already aggressively moving workloads into private and public cloud, as well as hybrid deployments.—Meredith Courtemanche
Containers Can Enable Real Agility

**IT managers, architects** and developers are all trying to come to terms with how microservices and containers will change business IT. It’s a good question on one level. But in reality, something more is at work.

To understand **microservices** and **containers**, start by grasping the flow of value this digital transformation defines; then align IT and data center capabilities with the transformative drivers. Finally, build for agility—not just for the next big thing.

IT planners and managers must understand that changes in the fundamental relationship between applications and workers—in particular, by the paradigm of contextual, event-driven, mobile workers—drive the use of containers and microservices.

The shift in direction will keep expensive, long-lived IT infrastructure aligned with a dynamic marketplace. It is not simply a matter of more efficient cloud hosting or lower operations complexity, but a matter of better serving organizations that demand IT responsiveness.

**WHERE WORKERS WORK**
Application software evolved from recording business activity to facilitating that activity. Applications built a path of least resistance for companies to follow in establishing operational plans. Enterprise architects have tried to align applications to business goals for more than a decade, but they’re often frustrated because the IT tools available dictate the way in which workers can conduct business.

The advent of smart mobile devices encapsulates this phenomenon. Workers who are empowered with mobile apps require support for a task, not a predefined workflow path to follow. The mobile worker is event-driven, which means applications must be as well.
Microservices aren’t an IT development looking for a business justification; they’re a way to structure apps to respond to tactical needs.

At the application level, microservices force architects and developers to think not only of features and processes as services, but to rethink the whole notion of workflow and application composition. With service-oriented architecture and other service-based methodologies, workflows bind components into composed applications. The goal of microservices is to let the worker’s activity bind and compose components. Each step and feature is made available on demand.

Perhaps the most critical aspect of microservices and agile IT is that every component is essential. In an app model where components are explicitly composed, you can meaningfully separate essential apps from those that aren’t. With microservices, a component could be part of a critical app, depending on the worker context. Compliance, security and reliability must meet standards overall, not just where needed.

The extemporaneous nature of microservices is what’s driving the container trend. Virtualization, be it in the data center or the cloud, imposes an overhead cost on the applications that depends on how it is mechanized. If services are small and tactical, virtualization overhead is difficult to justify. Containers are essential if microservices are to see wide adoption, and microservices are essential to an event-driven IT model.

Containers are needed for microservices to see wide adoption, and microservices are needed in an event-driven IT model.

Microservices must be based on the right component architectures, which allow IT to deploy and scale components dynamically. RESTful, stateless components and even functional programming will actively deter developers from building stateful components.

Development managers and architects now have tools, with vendors such as Microsoft and Oracle addressing functional programming evolution through Lambda expressions. However, they’ll have to impose functional discipline around microservices development.
because the imperative model of programming is still available.

**BEWARE OF THE TRAPS**

Adopting containers doesn’t guarantee that data centers will evolve optimally to support microservices and event-driven IT. The more agile application components become, the more inefficient their deployment could be.

Take steps to prevent that. Work with business stakeholders and developers so that the application design avoids excessive, unwarranted componentization. Runaway microservices use leads to IT inefficiency without improving end-user agility. Beyond this, the IT operations team must modify data center operations practices and the data center itself.

Microservices and event-driven components work best if network delays are minimized throughout the entire data center and even between multiple data centers or public cloud services. Use fast local area networks for the data center backbone, providing optimized network data paths in servers and network adapters. Avoid overloading virtual switches.

Interconnect data centers to reduce latency and packet loss.

What can’t be made faster should be avoided. Spec out new application deployment and DevOps tools as needed to place microservices properly. Host microservices close to the point of worker connection, but also as close as possible to fixed resources, such as databases. This should help avoid excessive propagation delay.

Seek out orchestration tools that consider these points when placing a microservice on IT resources. Train the IT operations staff to provide the correct connection information in installation parameters and policies.

Microservices and containers, driven as they are by business agility, will likely create the largest force that drives a data center to adopt private cloud, thus supporting hybrid cloud deployment. While it is possible to build a microservices and container model on virtualization principles, the need for agile application response to work events dictates open migration of features and functions. Cloud-friendly practices internally and with assiduous public cloud adoption will make elastic component scaling easier. —Tom Nolle
Deploying Docker on Bare Metal—Yes or No?

To some IT teams deploying containers into existing infrastructure, VMs offer compatibility with the rest of the environment and flexible deployment between hosts. Others argue that a Docker bare-metal deployment best fits with containers’ kernel-sharing efficiencies for better performance. Each camp argues that its approach is the more frugal option.

“For VMs, you buy a chassis with an ungodly amount of CPUs and a huge memory partition,” said Gary Davidson, senior architect for Vitacost.com, a division of Kroger Inc. that sells vitamins and supplements through an e-commerce site. This tends to be more expensive than groups of smaller, bare-metal servers, he said.

Vitacost is preparing to put containers into production later this year, and, for now, runs them on a VM infrastructure. Davidson said his ideal deployment choice is bare-metal servers.

A consultant working with Vitacost and others, however, said most of his clients prefer VMs over containers on bare metal.

“With VMs, you can quickly spin up images, and they can offer better resilience” with features such as high-availability failover and dynamic resource scheduling, said Vijay Ramanan, principal consultant with Lister Technologies, a multinational consultancy.

A Reality Check

Docker on bare metal might be the theoretical ideal, but it’s not the existing infrastructure at many companies, most of which are also not fully containerized yet, Ramanan said. “It is different if you’re talking about a 100% Docker infrastructure,” Davidson said.

Still, others contend that running bare-metal containers is a no-brainer.

“Why would you run it on a VM?” said a software engineer for a customer experience
management software maker on the West Coast who requested anonymity. “Docker does the bin packing for you and makes better use of the hardware.”

For companies using containers in the public cloud, VMs are unavoidable. Still, VMs’ scalability speeds and processes are somewhat antiquated.

For companies that want to use containers in hybrid or public cloud environments, VMs are unavoidable, said Pauly Comtois, vice president of DevOps for a multinational media company, a business that runs both Docker on bare metal and Docker VM deployments. Still, VMs’ scalability speeds and processes are somewhat antiquated, Comtois said, making bare metal the better option for some apps.

Docker itself isn’t picking sides. “When I came to Docker, everything in my life was VMs,” said Mike Coleman, a Docker technical evangelist with a focus on IT operations. Many companies approaching containers for the first time find themselves in that same situation, he said.

Coleman notes the difficulty in scaling out 1,000 containers in a web service if they all take too long to boot. That’s an argument for bare metal, he added.

Conversely, commercial software licensing costs are now tailored to virtualized environments and tend to charge by CPU core or socket, meaning VMs might be the less expensive way to go. VMs can also offer resource pools and quotas, areas where Docker can’t quite match them yet, Coleman said.

HOW TO THINK ABOUT HYPervisors
VMware resiliency features in particular, such as high availability, distributed resource scheduler and fault tolerance, “can be hard to live without,” Coleman said. “That might be a deciding factor for you.”

Additionally, organizations with mixed environments, whether running multiple Linux distributions or both Windows Server and Linux operating systems, can best mix workloads
using Docker on virtual machines.

Then again, hypervisors introduce latency for some applications. And while some applications are better consolidated using virtual servers, Coleman cited the example of Swisscom, which he said moved a MongoDB database deployment from a pool of over 400 VMs to a set of 20 hosts, each of which runs 20 containers.

Ultimately, Coleman stopped short of offering any direct guidance for those evaluating the choice between Docker bare metal and the VM.

“In the real world ... microservices and monoliths are a continuum,” he said. “And I don’t expect the way people get started [with containers] to be the way they’ll necessarily end up. We’re all in the early days here.”

—Beth Pariseau
MEREDITH COURTEMANCHE is the senior site editor for SearchITOperations.com.

TOM NOLLE is president of CIMI Corp., a strategic consulting firm specializing in telecommunications and data communications since 1982. He’s the publisher of Netwatcher, a journal addressing advanced telecommunications strategy issues.

BETH PARISEAU is senior news writer for TechTarget’s Data Center and Virtualization Media Group. Write to her at bpariseau@techtarget.com or follow @PariseauTT on Twitter.