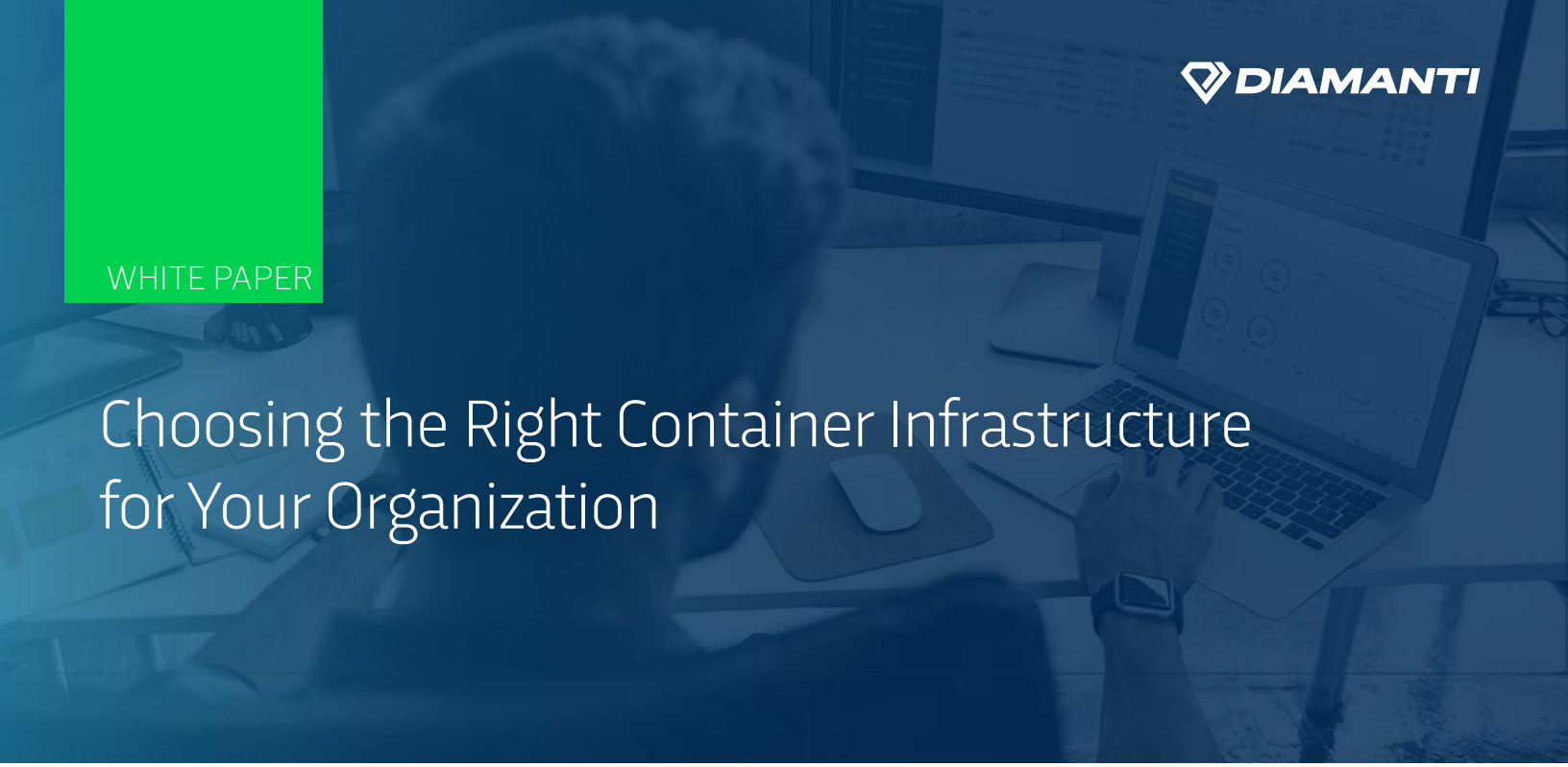

DIAMANTI

WHITE PAPER

Choosing the Right Container Infrastructure for Your Organization

Container adoption is accelerating rapidly. Gartner predicts that "by 2018 more than 50% of new workloads will be deployed into containers." Existing IT infrastructure has been optimized over a period of many years for virtualized business applications and may not efficiently support containers. As your organization navigates the transition from virtualization, you'll need infrastructure that addresses the unique needs of a container environment.

Given the increasing pressure on IT teams, you'll also want to consider carefully whether you will buy or build your infrastructure stack to support containers. As with any infrastructure decision, there are a number of factors to consider:

- Solution components: What components—hardware and software—make up the solution, and how well do they meet your requirements?
- Deployment time: How long will the solution take to deploy, and how much expertise will that require? Solutions that exceed your team's skills may need expensive professional services engagements, adding cost and time.
- Management: How much time and expertise will the various hardware and software components require to manage?
- Scalability: How difficult is it to scale the solution as your needs grow?
- Total cost of ownership (TCO): How much will it cost you to own and operate the solution, including staff time?

"We expect 30x growth of containerized apps in two years."
–Docker

"In 2018, more than 50% of new workloads will be deployed into containers."
–Gartner

In container environments, there are a few additional factors to consider:

- Bare-metal or virtualized? Bare metal is the gold standard for production containers. Running containers inside VMs adds an additional layer to the stack that must be managed and debugged, adding cost and complexity.
- Persistent storage: Applications running inside containers need to be able to save data permanently. Container solutions must provide a mechanism for persistent storage, even as containers come and go.
- Networking model: Getting networking right remains one of most difficult aspects of container environments, and container networkingmust integrate smoothly with your existing data center networking.
- Orchestration: Dynamic container environments require orchestration tools to coordinate activities and automate operations. While there are many options, Kubernetes has emerged as the clear leader.
- Support: How will you get support for the full infrastructure stack, including both hardware and software? From a single vendor? From multiple vendors? From the open-source community?

This paper examines the pros and cons of the available solutions for running containers on-premises.

	DIY	Virtualized Solutions	Diamanti
Bare metal?	√		√
Full stack integration?			√
Full stack 24×7 support?			√
Integrated orchestration?			√
Integrated networking?		?	√
Real-time IO service levels			√
IO performance	Variable	Low	High
Infrastructure efficiency	Low to Medium	Low	High
Time to deploy?	Weeks	Days	Minutes
Management complexity?	High	Medium	Low
TCO	High	Medium	Low

DO IT YOURSELF

The biggest advantage of a do-it-yourself container solution may also be its biggest disadvantage: complete freedom to choose all the hardware and software components.

- Do you want servers with only internal storage, or servers plus separate storage arrays?
- Bare-metal or virtualized?
- What version of Linux?
- Docker or a different container environment?
- What orchestration platform?
- Open-source or packaged software distributions?
- What other software (drivers, etc.) is needed for everything to work together?
- How will the resulting solution integrate with existing data center infrastructure?
- How will you get support?

These questions aren't impossible to answer, but if your team is new to container technology—or just needs to get a project off the ground quickly—there can be a lot to wrestle with. In today's rapidly evolving IT environment, the question of buy versus build has become a critical one.

If you choose to build out infrastructure yourself, deployment will take longer and may require professional services to complete. Ongoing management complexity will be higher than it would be with a more integrated solution since you'll have to keep up with the open-source community and integrate patches and enhancements as needed.

If you opt for servers with internal storage, scaling is straightforward. You simply add another server when resources run low. But you'll also have to provide a mechanism for data management and data protection on each server, figure out how to balance storage use across the set of servers, and you may need to provide a mechanism for shared storage such as NFS or a clustered file system such as Ceph or GlusterFS.

If you choose separate storage, it can simplify storage management initially, but scaling becomes more complicated. Suppose you start with a few servers and one storage array. To scale, you add servers until the storage array runs out of performance or capacity. Then you either add

PROS

- Complete flexibility

CONS

- Longer time to deploy
- Management overhead
- Lack of full-stack support
- Must keep up with community
- Professional services may be needed
- Solution complexity
- Solution resiliency
- High total costs due to management/staff overhead

The risk of "intellectual" lock-in

IT teams that take a do-it-yourself approach to container infrastructure may not always recognize the added Day 2 risks they face. Do-it-yourself infrastructure is like getting a puppy. The real work begins with the care and feeding.

Let's say you pull together a talented team for Day 1 and they create a solution tailored to your exact needs. But, over time, team members drift off to new positions or new companies. You may be left without the necessary skills or institutional knowledge to make configuration changes, software updates, or enhancements.

a second storage system or replace the original system with a more powerful one. Either way, this can be disruptive and can result in a big, and often unanticipated, incremental expense.

When it comes to networking, you're completely on your own. You need to make sure that your chosen networking model is compatible with your container and orchestration solutions. Again, that may not sound difficult, but the networking documentation for technologies like Docker and Kubernetes run to hundreds of pages, suggesting there's a lot to think about and plan for.

CONVERGED AND HYPERCONVERGED INFRASTRUCTURE SOLUTIONS

A variety of vendors have created converged and hyperconverged infrastructure solutions to reduce the complexity of IT infrastructure deployment:

- Converged infrastructure (CI) pre-packages several servers with a separate storage array.
- Hyperconverged infrastructure (HCI) combines servers with internal storage, software to virtualize that storage, and virtualization software such as VMware vSphere.

These solutions can be deployed for use in container environments. However, as a class, they are designed for virtualization rather than containers, making bare-metal container deployment impossible in almost all cases. The vendors themselves remain largely focused on virtualization. Therefore, the level of support you'll get from a vendor for everything in the infrastructure stack above virtualization is going to be minimal.

You may have to rely on the open-source community for container and orchestration support.

Why is bare metal the gold standard for containers?

Bare-metal containers provide optimal performance, allowing applications to access hardware without the need for pass-through or hardware emulation.

Bare-metal delivers many of the perceived advantages of virtualization including application portability and isolation. Running containers inside virtual machines is like doing virtualization on top of virtualization, it's unnecessary.

SOLUTIONS BASED ON VMWARE VSPHERE

A variety of CI and HCI solutions are available that have been optimized for VMware vSphere. By far the best known of these are the VxRail, VxBlock, and VxRack solutions from Dell EMC. These solutions simplify hardware deployment, but require you to run your container environment on top of the ESXi hypervisor.

Once the hardware is deployed, you have three official paths to deploy containers:

- vSphere Integrated Containers (VIC): VIC provides a Docker-compatible container engine, a container management portal, and a container registry, allowing containers to run in VMs along with other virtualized workloads. It requires vSphere vCenter.
- Photon Controller: Photon Controller is an open source project from VMware that allows you to cluster a number of hosts running the VMware ESXi hypervisor exclusively for the purpose of running containers. It lets you have your choice of orchestration frameworks and does not require VMware vCenter.
- Pivotal Container Service (PKS): Announced at VMworld in 2017, PKS combines Pivotal Cloud Foundry with Kubernetes and the Harbor container registry, all running on top of VMware vSphere and VSAN. It was due to be available in the last quarter of 2017. PKS is envisioned as a validated solution, meaning installation of the software components remains a do-it-yourself effort.

To deploy a container solution on VMware, you have a number of possible options. Deploying the software stack and ongoing management will consume some added time because of the number of software components involved and the lack of integration out of the box. After initial deployment, scaling is relatively straightforward for HCI since you simply deploy additional VMs configured with the full software stack on new servers as they are added. With CI, you face the challenge of scaling storage separately, a larger incremental cost.

Persistent storage is provided by:

- VMware vSAN software-defined storage on VxRail
- ScaleIO software-defined storage on VxRack
- VMAX storage for VxBlock

PROS

- Multiple consumption options

CONS

- No bare-metal deployment
- VMware licensing costs and higher management overhead drive up TCO
- No full-stack support
- Bewildering range of choices
- Complex software deployment
- Virtualization plus network limitations reduce container density

The vSphere Docker Volume Service is needed to allow Docker containers to consume vSphere storage.

Container networking in VMware environments relies on VMware networking mechanisms. Many who have deployed containers on VMware have found this to be the Achilles' heel of the solution. The worst-case scenario results in deployment of a single container per VM, wasting resources and eliminating the density advantage of containers.

NUTANIX ENTERPRISE CLOUD

Nutanix is the best-known and most widely deployed HCI solution other than the Dell EMC options just described. Nutanix hardware arrives pre-installed with the virtualization stack, making the hardware straightforward to deploy. That leaves a number of software installation and configuration steps to get containers running on the platform, including installation of the latest version of Nutanix Acropolis Container Services (ACS) and installation of the Docker ecosystem on top of virtual machines running Linux.

Persistent storage requires the Nutanix Docker Volume Plugin from the Docker store. Once installed, the plugin works with ACS to connect to persistent storage volumes.

Once everything has been installed, the container environment can be monitored and managed using the Nutanix Prism management tool. Scaling is achieved by adding additional nodes as more compute and storage capacity are needed.

Container orchestration in the Nutanix environment remains a moving target at this time. While it is possible to implement container orchestration, it remains a do-it-yourself project for now. Nutanix has announced the intention to support Kubernetes in association with Google beginning in 2018.

As with VMware solutions, you are beholden to Nutanix networking capabilities for your container environment. Nutanix best practices for containers so far have little to say on the subject of networking.

PROS

- AHV eliminates VMware licensing costs

CONS

- No bare-metal deployment
- No full-stack support
- Virtualization reduces container density
- Complex software deployment and management, adds to TCO
- Orchestration support unclear

DIAMANTI HYPERCONVERGED CONTAINER PLATFORM

Diamanti has created the first infrastructure platform purpose-built for containerized applications, combining the ease of use of hyperconverged infrastructure with the unparalleled performance and efficiency of bare-metal containers.

Diamanti has the only turnkey solution specifically built for container infrastructure on the market. We bring the ease-of-use of the cloud to your on-premises container deployment, and add capabilities you won't find in the cloud or anywhere else.

Efficient, shared infrastructure for containers provides persistent storage and plug-and-play networking that integrates with existing data center technologies. The result is a highly-available pool of CPU, memory, network, and storage resources delivered to containers on-demand, with full QoS for all resources including storage and networking, something no other vendor offers.

The Diamanti platform integrates everything—hardware and software—out of the box, so it can be fully deployed and operational in minutes. You'll be able to start running containerized applications immediately, without having to spend weeks or months standing up a DIY solution. Open-source software, including Docker, CentOS, and Kubernetes, is pre-installed and ready to run containers, so there's no vendor lock-in. The environment can be managed by anyone familiar with these open-source tools; an intuitive UI makes managing and monitoring the platform simple, even for those without prior infrastructure management experience.

Scaling occurs through the addition of nodes to a cluster. Because containers run on bare-metal, container density is extremely high; hardware utilization approaches 90%.

Diamanti is the only infrastructure solution that delivers additional value to the container environment out of the box. A built-in converged I/O controller virtualizes network and storage for containers, guaranteeing application performance without code changes or customization. The Diamanti architecture delivers order-of-magnitude latency improvements compared with traditional shared-storage systems and software overlays.

PROS

- Full-stack integration
- Full-stack, 24×7 support
- Fast to deploy
- Easy to manage
- Easy to scale
- Eliminates vendor lock-in
- Highest container density
- High resource utilization
- Ultra-low latency

CONS

- May have higher capex than some DIY options

Only Diamanti offers:

- Seconds to deploy applications with guaranteed performance
- 10x higher performance (1 million+ IOPS) vs. traditional solutions, with low latency and fast throughput
- No time spent configuring infrastructure for containers
- 6x improvement in infrastructure utilization vs. traditional solutions

MAKING SMARTER INFRASTRUCTURE CHOICES

The need to provide new digital services is forcing enterprises to pivot to container infrastructure, similar to the pivot to virtualization a decade ago. As you make this transition, you have to consider your requirements and make careful infrastructure choices to avoid being saddled with a solution that is overly complex, difficult to manage, that lacks the necessary performance, or that locks you into a specific vendor environment.

Only Diamanti satisfies these needs. Full-stack integration and full-stack support allows developers and operators to avoid vendor lock-in while taking full advantage of familiar container software including Docker and Kubernetes. Storage and networking QoS deliver the performance necessary for demanding production environments. Diamanti requires no code changes. Using services that are already supported by standard Linux removes the need for custom protocols and drivers.

For more information, visit www.diamanti.com. To request a demo, email demo@diamanti.com

Diamanti, Inc. 111 N Market St., Suite 800, San Jose, CA 95113
(408) 645-5111