



Pave The Way To Cloud Native Application Success With The Right Infrastructure Approach

Wallie Leung, Sr. Dir. Solution Engineering

Agenda

- Enterprises are going Cloud Native
 - What attributes define ‘cloud native’?
- What Options Do I Have To Deploy My Cloud Native Apps?
- The DIY Docker/Kubernetes Stack In Depth
- Why You Should Run Containers On Bare Metal, And Not VMs
- A Look At Container Persistent Storage And Container Networking
- Case Study
 - Fortune 50 Bank Adopts Containerized DBs, Deploys On Bare Metal Infrastructure

Enterprises are Going ‘Cloud Native’: A Look Back At 2018

80 BILLION

Docker downloads to date



More than 50%
of new workloads will be deployed
in containers in 2018

Gartner

>70% of enterprises
plan to deploy containers in
production in 2018*

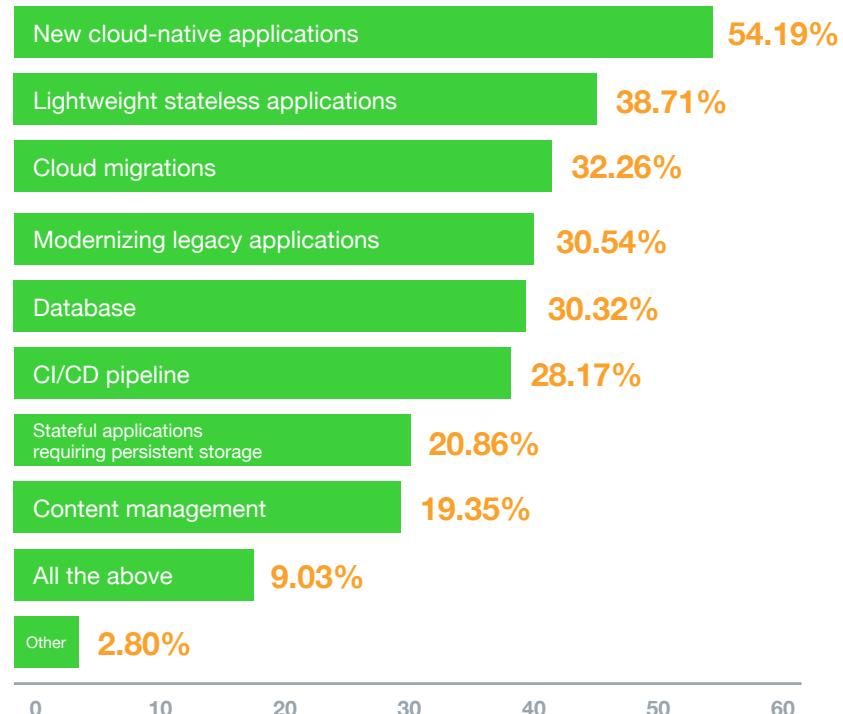
DIAMANTI

31% are modernizing
legacy apps with containers*

DIAMANTI

Diamanti Survey

How are containers being used?



What is 'Cloud Native'? 10 Key Attributes

1. Packaged as lightweight containers
2. Developed with best-of-breed languages and frameworks
3. Designed as loosely coupled microservices
4. Centered around APIs for interaction and collaboration
5. Architected with a clean separation of stateless and stateful services
6. **Isolated from server and operating system dependencies**
7. **Deployed on self-service, elastic, cloud infrastructure**
8. Managed through agile DevOps processes
9. **Automated capabilities**
10. **Defined, policy-driven resource allocation**

Source: <https://thenewstack.io/10-key-attributes-of-cloud-native-applications/>

What Options Do I Have for Cloud Native App Deployments?

Public Cloud

PROS:

- 100% OPEX means lowest cost at small scale
- Proven, resilient infrastructure as a service

CONS:

- Highest costs at large scale
- Difficult to optimize efficiency
- Security and compliance concerns
- No bare metal support yet

On-Prem: DIY with Hypervisor

PROS:

- Low complexity at small scale
- Experience with traditional technology components

CONS:

- Longer time-to-value
- With a VM layer, resource efficiency is reduced and complexity increases
- High complexity and cost at scale
- Difficult to manage at scale

On-Prem: DIY with Bare Metal

PROS:

- Flexible, seamlessly scalable infrastructure
- Better overall performance at scale
- TCO advantages are realized

CONS:

- Specialized skill sets required at different areas of the stack

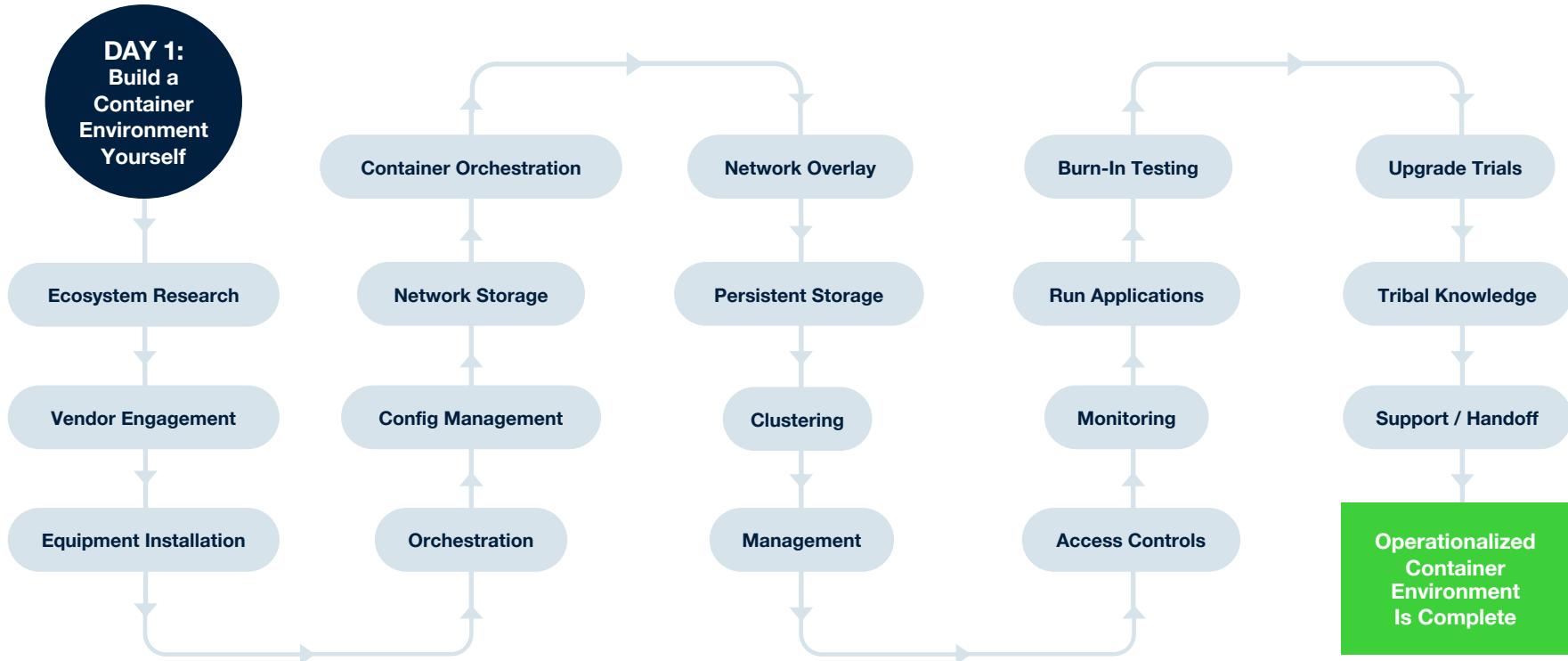
Infrastructure Remains a Top Container Adoption Challenge

“Enterprise interest in Kubernetes to build and deploy new applications is off the charts. **Security, storage, networking and monitoring are the top challenges** that our user community have highlighted on the Kubernetes adoption path.”

-- Dan Kohn, Executive Director



Do-It-Yourself Approach to Container Infrastructure



CNCF Landscape

CNCF Cloud Native Landscape

2019-03-05T13:25:14Z e5225b3

Database



Streaming & Messaging



See the interactive landscape at l.cncf.io

Application Definition & Image Build



Continuous Integration & Delivery



Platform



Observability and Analysis



Greyed logos are not open source

Scheduling & Orchestration

Coordination & Service Discovery

Remote Procedure Call

Service Proxy

API Gateway

Service Mesh

Orchestration & Management



Certified Kubernetes - Hosted



Logging



Runtime



Tracing



Provisioning



Container Registry



Chaos Engineering



Automation & Configuration

Container Registry

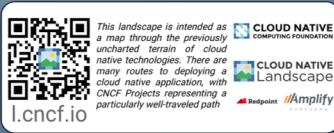
Key Management

Public

Kubernetes Certified Service Provider

Serverless

Cloud



Special



Major Container Adoption Challenges



Day 1 (Hard)

Deploy infrastructure

- Build the container stack
- Configure network, storage
- Install container runtime and orchestration



Day 2 (Harder)

Manage containers in production

- Guaranteed real-time SLAs
- Infrastructure services
- 24x7 full stack support



Day 3 (Hardest)

Expand with multi-cloud

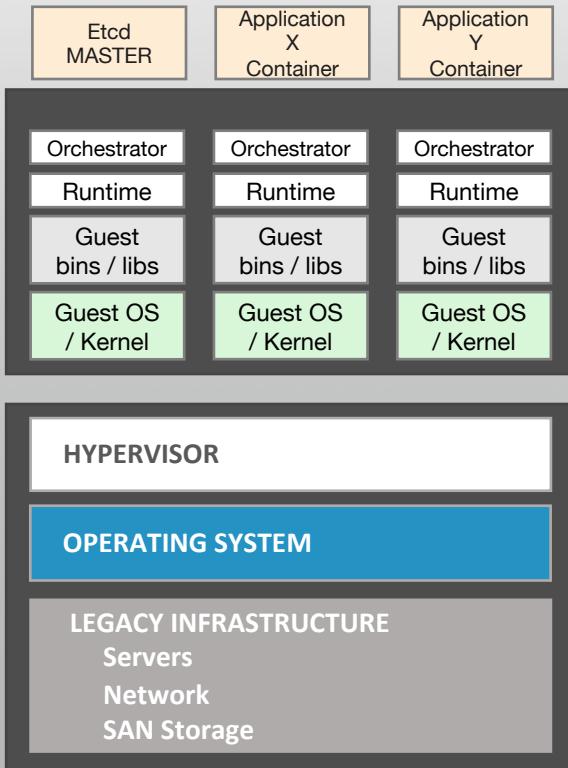
- Quick movement of containers across cloud environments
- Seamless scalability
- Policy-driven

**Choose your cloud native infrastructure wisely;
it matters at every step.**

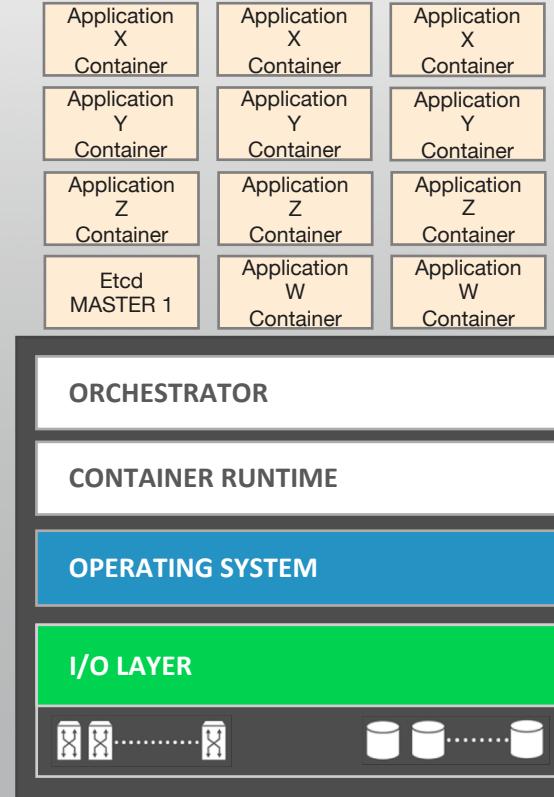
Containers: Benefits Beyond Resource Utilization

	Containers	Virtual Machines
Configuration Management	<ul style="list-style-type: none">Changes are made directly to the container image	<ul style="list-style-type: none">Requires external CM engines or hypervisor snapshot functions
Scalability	<ul style="list-style-type: none">Containers deploy in seconds;service components can be deployed independently	<ul style="list-style-type: none">Virtual images are entire host systems (>10 GB)Not very portable
High Availability	<ul style="list-style-type: none">K8s services can have defined resiliency attributesK8s works to ensure service uptime requirements	<ul style="list-style-type: none">seen from a monolithic perspectiveheavy backend investments to ensure uptime
Complexity	<ul style="list-style-type: none">Isolated, lightweight environments have fewer moving partsOps teams can focus on apps and business needs	<ul style="list-style-type: none">Many moving partsguest VMs have to emulate the full hardware stack of a server

Infrastructure: VMs VS. Bare Metal Hyperconverged



- Complex management
- Inefficient resource utilization
- Low container density
- Limited performance
- High TCO





**Container Persistent Storage
and Container Networking**

Drilling Down: Storage For Stateful Containers

Local Storage

- Host paths can be mounted to containers in Docker and Kubernetes

CHALLENGES

- Persistence
- High Availability (HA)
- Who is managing it?

Traditional SAN Storage

- Either connect by iSCSI or FC

CHALLENGES

- Persistence

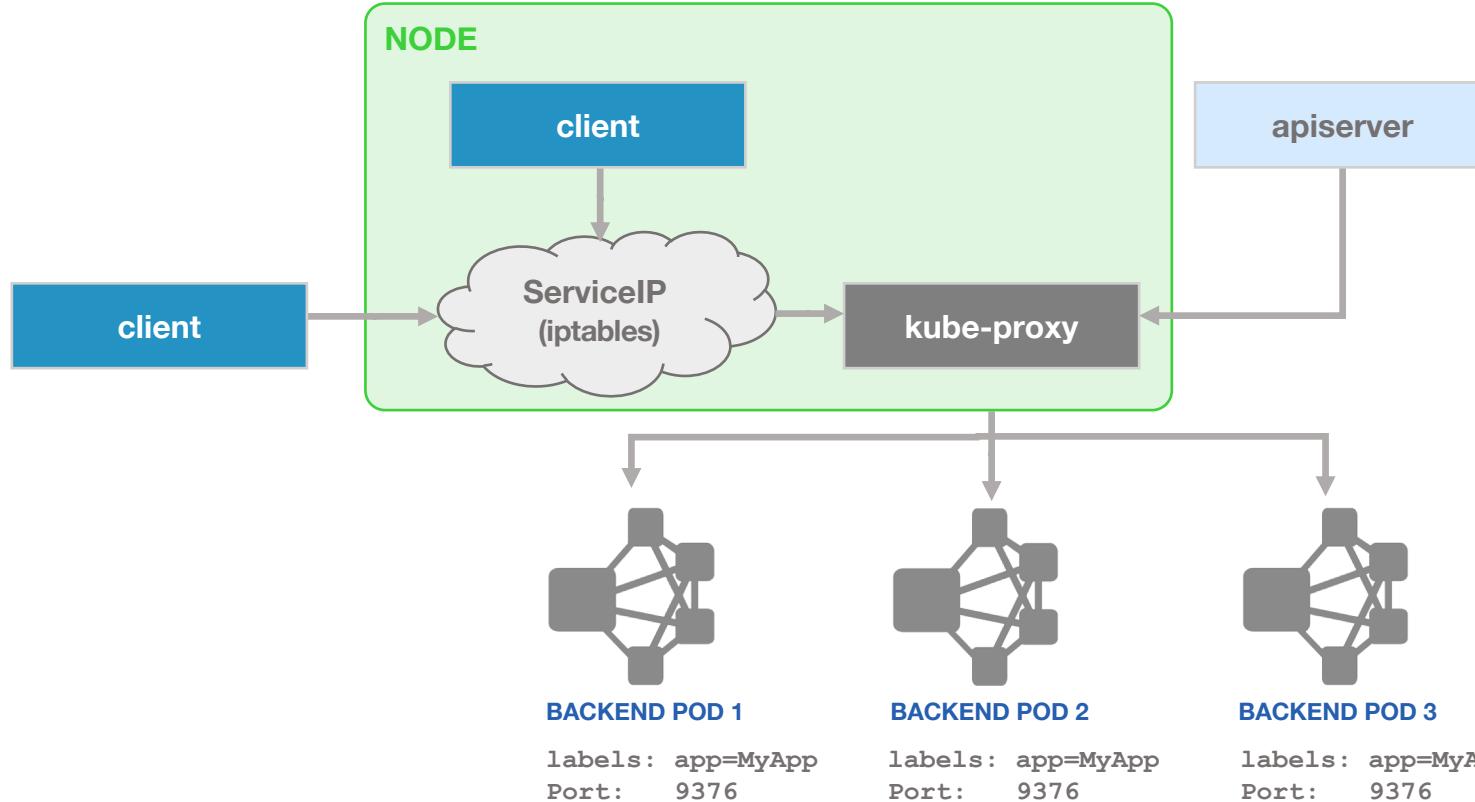
Network File System (NFS)

- Offers persistent storage

CHALLENGES

- Performance isn't adequate

Drilling Down: Kubernetes Networking Model



Source: <https://kubernetes.io/docs/concepts/services-networking/service/>



Case Study: Database-as-a-Service on Bare Metal Infrastructure

Fortune 50 Bank: Migration to Containerized DBaaS

Profile:

- Large global bank serving 16M Canadian, American, and international customers
- 81,000 employees, worldwide

Challenge:

- Migration away from Oracle for cost, agility
- ~4 weeks to deploy new DBs on traditional infrastructure

Initial approach:

- DIY infrastructure to support containerized DBaaS
- \$1.2M HW/SW investment
- 48 nodes & 4 dedicated FTEs

Minimal Downtime

16x
Footprint reduction

23x
Faster deployment

\$14M
TCO savings over 3 years

Solution:

- Installed and configured 3-node Diamanti D10 cluster
- Deployed PostgreSQL, MongoDB, SQL Server and MariaDB on Diamanti in minutes, vs. weeks
- Saved \$1.2M in Oracle license costs
- Added D10 nodes dynamically with zero service disruption

Intel NVMe Drives Multi-Workload Efficiency

End-User Value



Breakthrough performance
IOPS

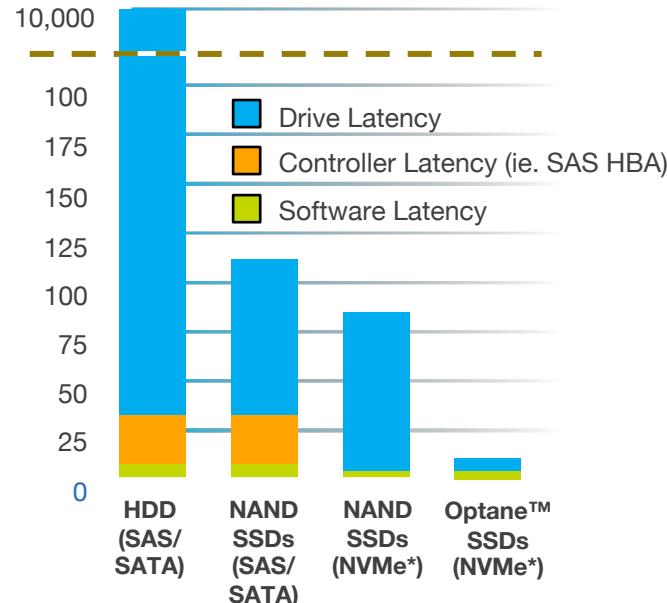


Low latency



Predictably fast service
QoS

Latency (μs)



The latency comparisons above include internal Intel estimates for the SSD NAND NVMe* and SSD with Intel® Optane™ technology



*Other names and brands may be claimed as the property of others

Diamanti Bare-Metal Kubernetes Platform

Complete turnkey Kubernetes stack

- Hyperconverged 1U appliance built on x86 architecture
- Features container-optimized networking and storage models
- 24/7 full-stack support

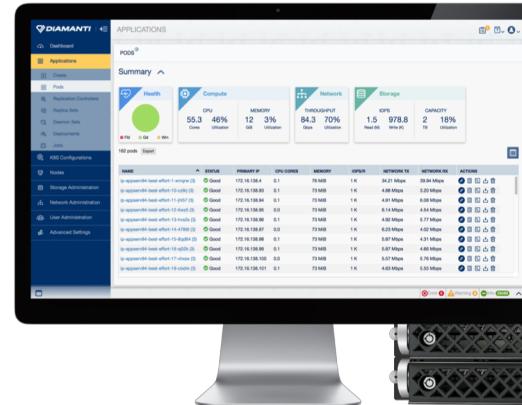
Built for public cloud experience, on-prem

- Manage multiple on-prem clusters and private-cloud deployments through single UI
- Per container network and storage QoS
- Enterprise DP/DR features: mirroring/synchronous replication, snapshots/asynchronous replication
- Burst production workloads to the cloud
- Dynamic service management through Istio

Benefits:

- High performance
- Efficient
- Secure
- Installs in minutes
- Low TCO

Database Partners:





Thank You