



OpenShift Hyper-Converged Infrastructure Bare Metal Deployment with Containerized Gluster

Greg Hoelzer
Sr. Container Application & Middleware Solution Architect
January 2017 – Minneapolis Red Hat Users Group

Question -

If I'm building a Container Application Platform from the ground up, do I need Virtualization or Specialized Storage?



The Problem -

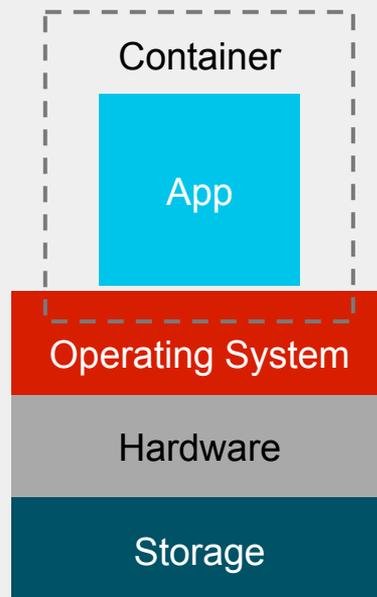
Different Applications and Services can have varying Workloads, Integration, and Persistence Requirements creating Complex Infrastructure



SIMPLIFIED SOLUTION – NOT Virtualization

Deliver Infrastructure AND Application Services using Containers and Standardized/Commodity Hardware Configurations with increased Automation

Controlled by IT Operations

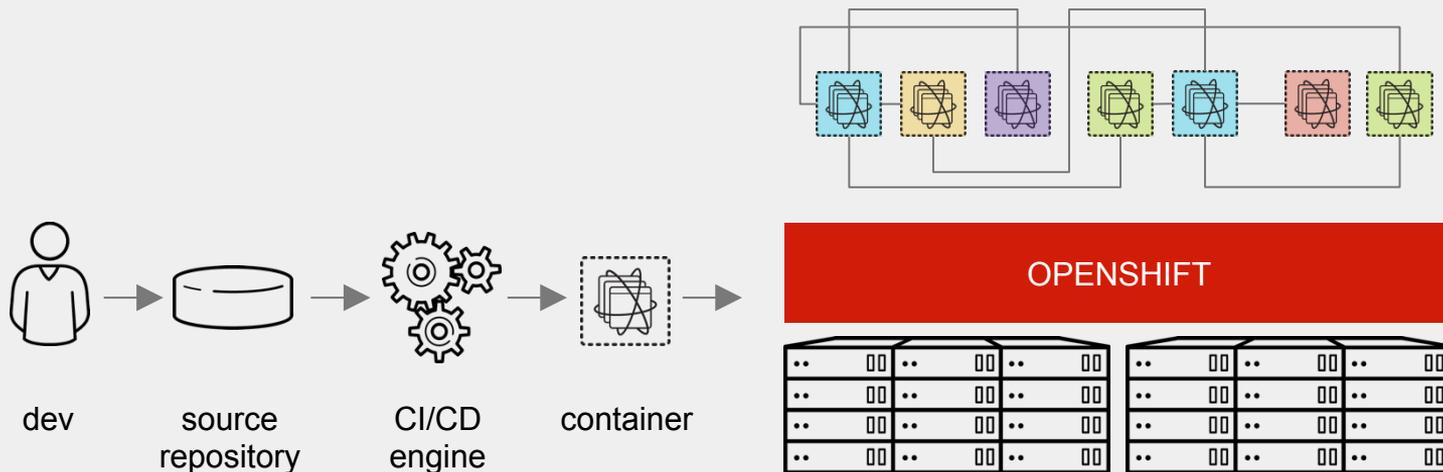


Controlled by Developers

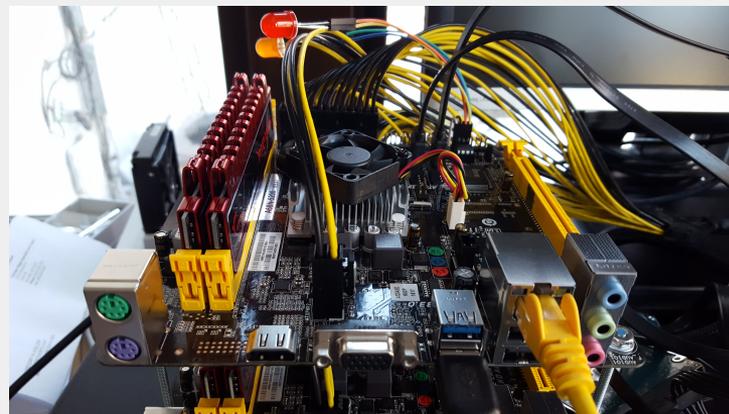
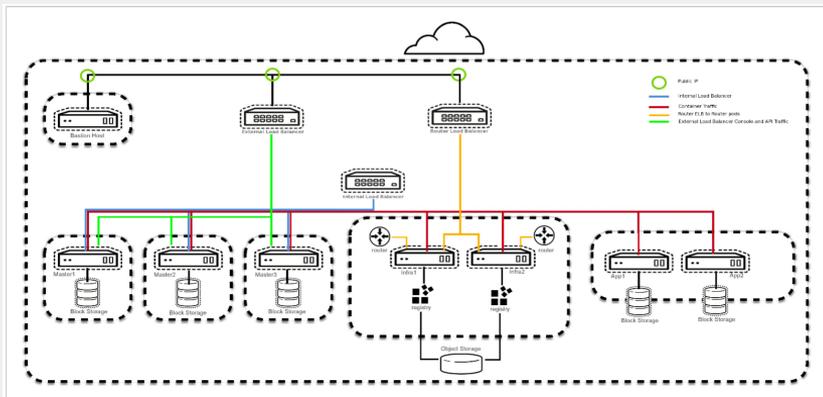


Orchestrated by Kubernetes

DELIVERED WITH OPENSSHIFT



LET'S BUILD IT – KONTAINER GARDEN 1.0



Hosts - 4 Core AMD/Mini-ITX with 8 GB RAM

1 Master + Infra Services Host

- RHEL Enterprise
- 500 GB/7200 RPM Boot + Data Volume
- 128 GB/SSD Docker Volume

2 Container Node Hosts

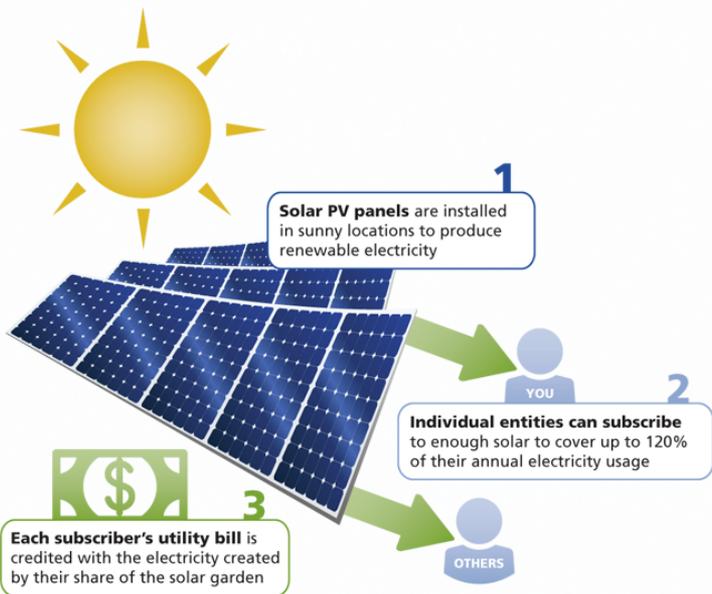
- RHEL Atomic
- 128 GB/SSD Boot + Docker Volume
- 500 GB/7200 RPM Data Volume

“Kontainer Garden” – Where did that come from?

Solar Garden

Community solar gardens are for people that want to go solar but are unable to do so on their own.

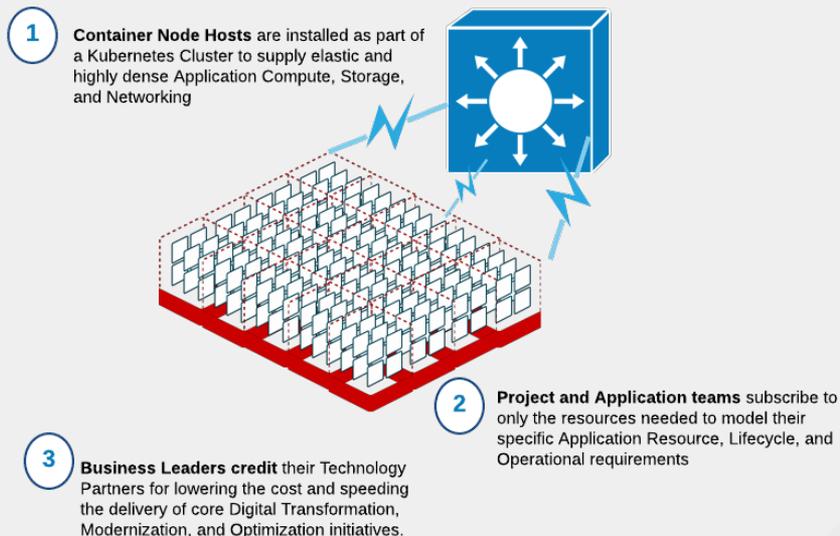
Perhaps you live in an apartment, have a shaded roof at home, or don't have space at your organization. Now you can subscribe to a community solar garden installed near you and get credits on your utility bill.



Kontainer Garden

Open Source Community-Driven Kontainer Gardens are for Enterprises that wish to leverage Docker Containers with Kubernetes Orchestration and want a partner to support a complete Container Application Platform

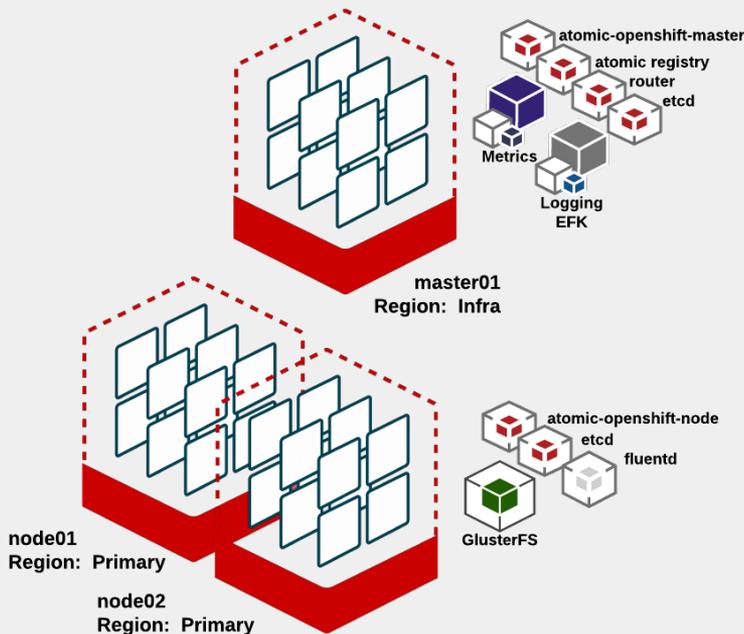
Businesses are being pushed grow revenue through Digital Transformation while optimizing operations and modernizing existing technology Infrastructure and accelerating delivery timelines



LET'S INSTALL AND CONFIGURE IT



Kontainer Garden 1.0 3.3 Cluster Configuration



MVP representative of HCI for Containers

Master Host

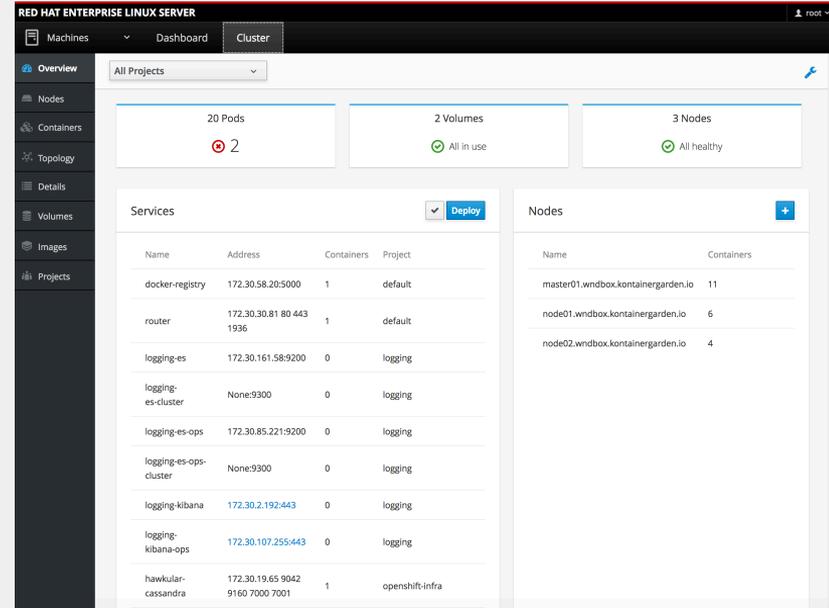
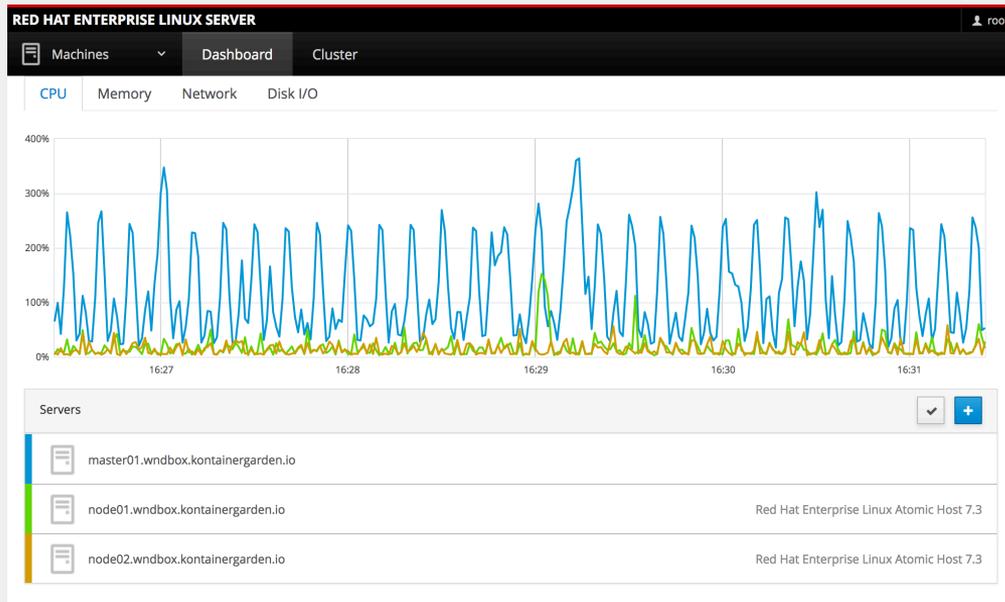
- Atomic OpenShift Master + etcd
- Infrastructure Services (Registry/Router)
- Metrics Services (Heapster/Hawkular)
- Logging Services (EFK Stack)
- NFS Storage (Registry/Elastic Search)

Node Hosts

- Atomic OpenShift Node + etcd
- Logging Services (Fluentd)
- Containerized Storage (GlusterFS)
- *Application Container Workloads*

LET'S MONITOR AND MANAGE IT

Using Cockpit for both RHEL Enterprise and Atomic Hosts
- For system/machine and OpenShift Cluster



ADD CONTAINERIZED STORAGE TO IT

Using a GlusterFS Daemon Set and Kubernetes Service/External Endpoints - For Test/Replica 2 Storage Cluster

The screenshot displays the OpenShift console interface for a project named 'gluster'. The left sidebar contains navigation options: Overview, Applications, Builds, Resources, Storage, and Monitoring. The main content area shows the 'glusterfs-cluster' configuration, indicating 'No deployments' and 'There are no deployments or pods for service glusterfs-cluster.'

Below the configuration, two pod details are shown for 'gluster-3k1x1' and 'gluster-k403l', both running on 'gluster/rhgs-server-rhel7'. Each pod's resource usage is visualized with a bar chart and a circular gauge showing '1 pod'.

- Pod gluster-3k1x1** (13 days ago):
 - 234.2 MiB Memory
 - 0 Millicores CPU
 - 0 KiB/s Network
- Pod gluster-k403l** (13 days ago):
 - 207.7 MiB Memory
 - 1 Millicores CPU
 - 0 KiB/s Network

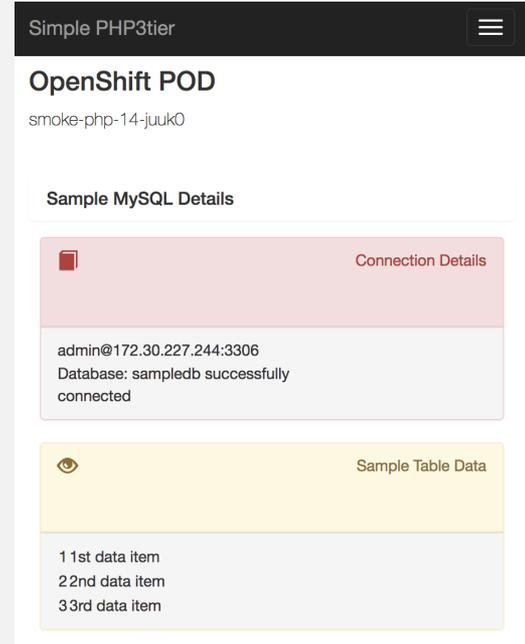
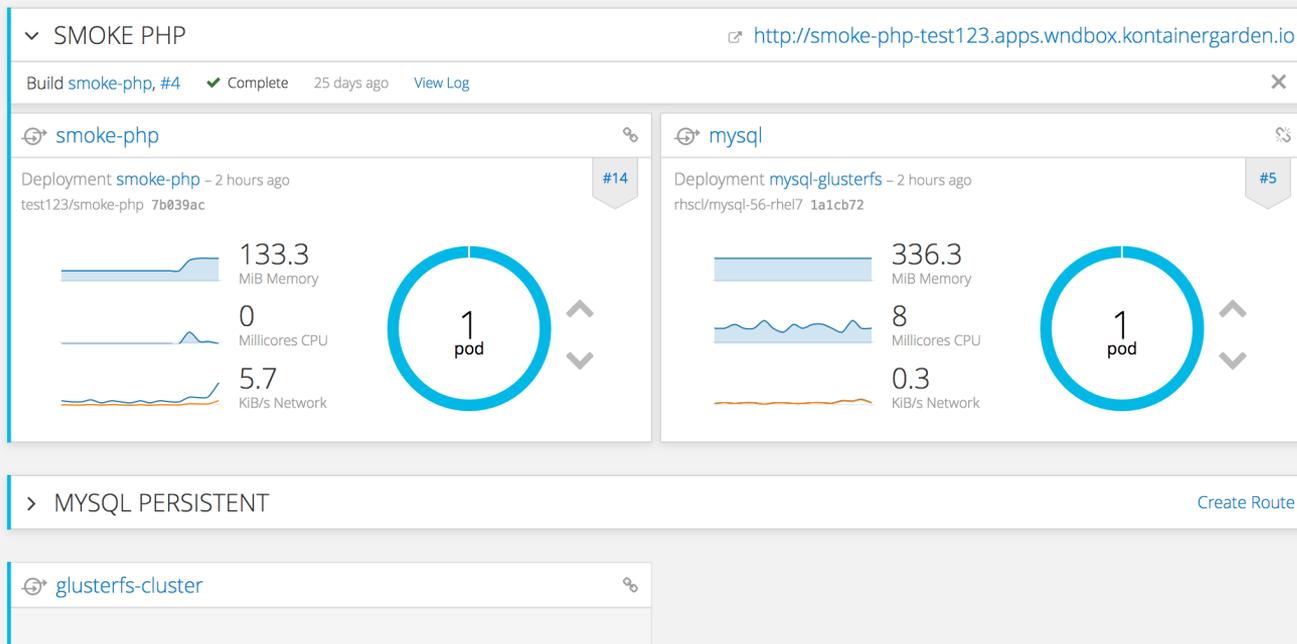
On the right, a terminal window shows the command 'gluster' and its output for 'gluster volume info testvolume':

```
sh-4.2# gluster volume info testvolume
Volume Name: testvolume
Type: Replicate
Volume ID: 1400ff34-2465-4b82-9094-bbec67bd40d5
Status: Started
Number of Bricks: 1 x 2 = 2
Transport-type: tcp
Bricks:
Brick1: 192.168.0.110:/mnt/brick1/testvolume
Brick2: 192.168.0.111:/mnt/brick1/testvolume
Options Reconfigured:
performance.readdir-ahead: on
sh-4.2#
sh-4.2# gluster volume status
Status of volume: testvolume
Gluster process                                TCP Port  RDMA Port  Online  Pid
-----
Brick 192.168.0.110:/mnt/brick1/testvolume     49152      0          Y       297
Brick 192.168.0.111:/mnt/brick1/testvolume     49152      0          Y       288
NFS Server on localhost                        2049      0          Y       282
Self-heal Daemon on localhost                 N/A       N/A        Y       291
NFS Server on node02.wndbox.kontainergarden.io 2049      0          Y       276
Self-heal Daemon on node02.wndbox.kontainergarden.io N/A       N/A        Y       283

Task Status of Volume testvolume
-----
There are no active volume tasks
```

SMOKE TEST IT

Using basic PHP/MySQL Application - With Gluster Persistent Volume & Claim



HARDEN AND TUNE IT

MASTER

```
#swapoff -a
```

```
#vi node-config.yaml
```

```
kubeletArguments:
```

```
image-gc-high-threshold:
```

```
- "80"
```

```
image-gc-low-threshold:
```

```
- "60"
```

```
kube-reserved:
```

```
- "cpu=300m,memory=300M"
```

```
system-reserved:
```

```
- "cpu=200m,memory=200M"
```

```
max-pods:
```

```
- "20"
```

```
serialize-image-pulls:
```

```
- "false"
```

NODES:

```
#swapoff -a
```

```
#vi node-config.yaml
```

```
kubeletArguments:
```

```
cpu-cfs-quota:
```

```
- "false"
```

```
eviction-hard:
```

```
- "memory.available<500Mi"
```

```
image-gc-high-threshold:
```

```
- "80"
```

```
image-gc-low-threshold:
```

```
- "60"
```

```
kube-reserved:
```

```
- "cpu=300m,memory=300M"
```

```
system-reserved:
```

```
- "cpu=200m,memory=200M"
```

```
max-pods:
```

```
- "50"
```

```
serialize-image-pulls:
```

```
- "false"
```

Optimize Pod Scheduling & Performance

Enable CPU Overcommit

Eviction Strategy & Limit

Optimize Docker Image Garbage Collection

Reserve System Resources

Limits Pods based on Host Size & Function

Enable Parallel Docker Image Pulls

STABILITY TEST IT

Using basic PHP/MySQL Application (75 Pods) and Example .NET Core Application (20 Pods)

RED HAT ENTERPRISE LINUX SERVER

Machines Dashboard Cluster

Overview

All Projects

112 Pods 2 3 Volumes All in use 3 Nodes All healthy

Services Deploy

Name	Address	Containers	Project
docker-registry	172.30.58.20:5000	1	default
router	172.30.30.81 80 443 1936	1	default
glusterfs-cluster	172.30.88.198:1	0	gluster
logging-es	172.30.161.58:9200	0	logging
logging-es-cluster	None:9300	0	logging
logging-es-ops	172.30.85.221:9200	0	logging
logging-es-ops-cluster	None:9300	0	logging
logging-kibana	172.30.2.192:443	0	logging
logging-kibana-ops	172.30.107.255:443	0	logging
hawkular-cassandra	172.30.19.65 9042 9160 7000 7001	1	openshifts-infra

Nodes +

Name	Containers
master01.wndbox.kontainergarden.io	10
node01.wndbox.kontainergarden.io	50
node02.wndbox.kontainergarden.io	53

RED HAT ENTERPRISE LINUX SERVER

Machines Dashboard Cluster

CPU Memory Network Disk I/O

RED HAT ENTERPRISE LINUX SERVER

Machines Dashboard Cluster

CPU Memory Network Disk I/O

Servers

- master01.wndbox.kontainergarden.io
- node01.wndbox.kontainergarden.io
- node02.wndbox.kontainergarden.io

OBSERVATIONS AND IMPACTS

INFRASTRUCTURE

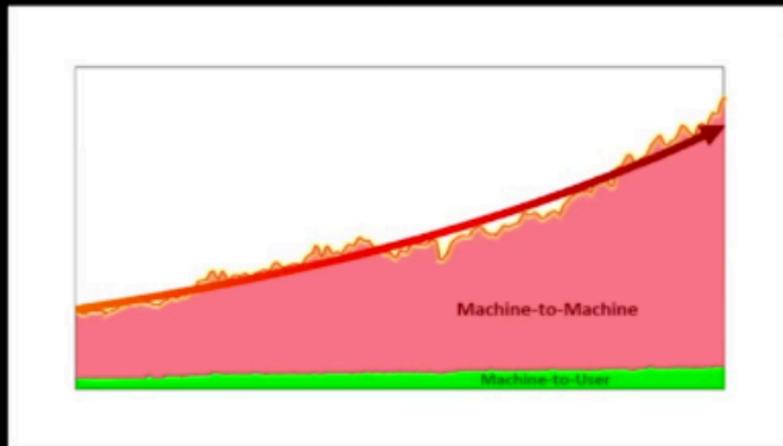
- RHEL Atomic OS now mature enough to seriously evaluate for running Containers
- Docker/Kubernetes together provide a compelling alternative to Virtualization
- Containerized/Converged Storage relatively easy to deploy, OpenShift 3.4 provides enhancements needed for Scale

APPLICATIONS

- Density/Over-Commitment of Resources can be tuned at the Platform and Application levels
- Both Scale-Out and Clustered Data Persistence possible with GlusterFS
- Master/Infrastructure Application Services may still benefit from Virtualization

LARGEST ORGANIZATIONS ARE SEEING?

Traffic growth: exponential



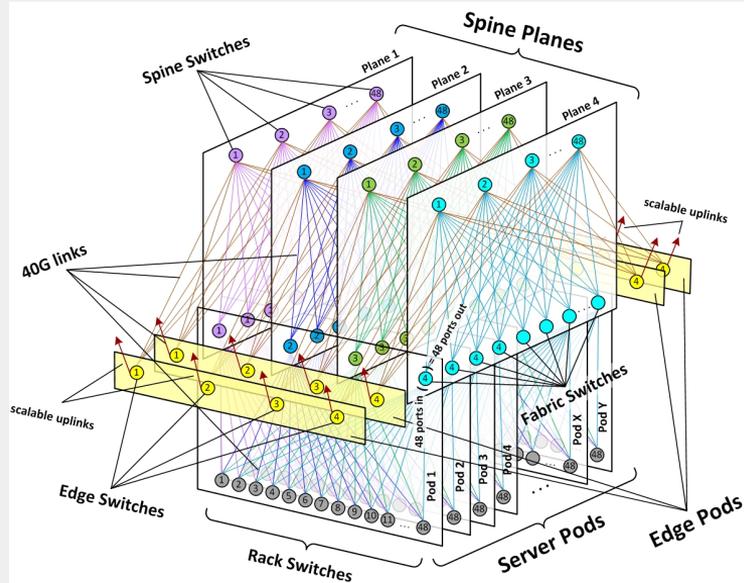
Machine to Machine:

Inter-Cluster Traffic

Machine to User:

Egress: out to Users/Internet

Network/Data Traffic Growth driving specialized physical Data Center, Switch, Server and Storage designs



<https://code.facebook.com/posts> - Introducing data center fabric, the next-generation Facebook data center network

START SMALL



TO GO BIG





THANK YOU



plus.google.com/+RedHat



facebook.com/redhatinc



linkedin.com/company/red-hat



twitter.com/RedHatNews



youtube.com/user/RedHatVideos