S111017- Implementing DevOps and Hybrid Cloud

Srihari Angaluri
Lenovo Data Center Group

Red Hat Summit 2017

5/3/2017

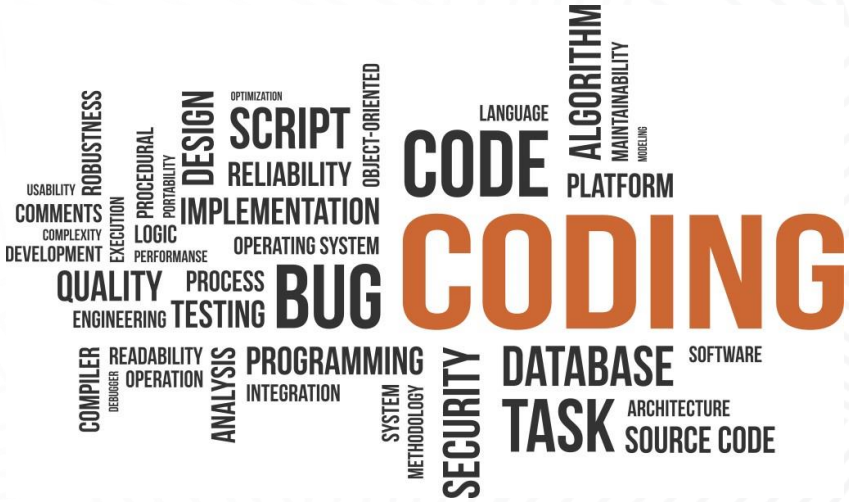
Outline

- DevOps and Containers
- Architectural Considerations
- Lenovo Cloud Technology Center
- Implementing Red Hat OpenShift
- Hybrid Cloud Management with CloudForms
- Lenovo Converged Platform
- Demo

Build Great Software – Fast…

Most Companies Have:

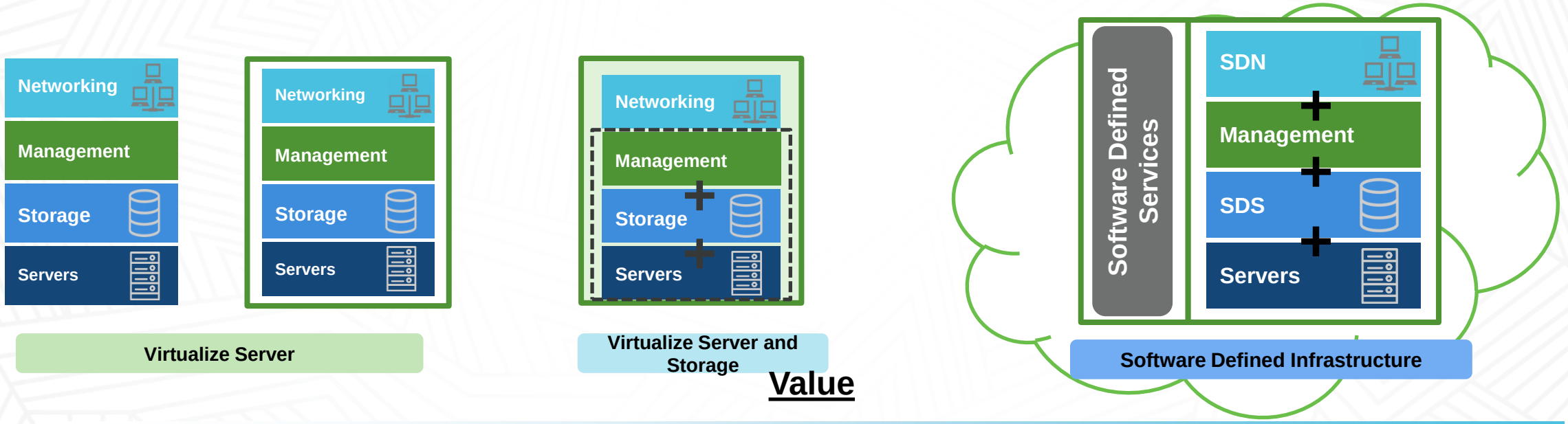
- A Lack of effective and modern processes & principals
- Average to poor inter-department collaboration
- Tools that are not interconnected in the workflow
- Lots of software re-coding “rollbacks”
- Infrastructure challenges; usually can’t get what they need – when they need it
- Control & governance concerns
- Issues they don’t know how to fix (in some cases know they have issues)



Industry Evolution

Traditional	**Wave1:** Converged	**Wave 2:** Hyper Converged	**Wave 3:** Software Defined Infrastructure (SDI)
Acquired as discrete parts Hard to deploy	Acquired as single entity Faster deployment	Moves storage into server	IT Agility - virtualizes server, storage, network, and public cloud resources

Applications / Workloads



Value

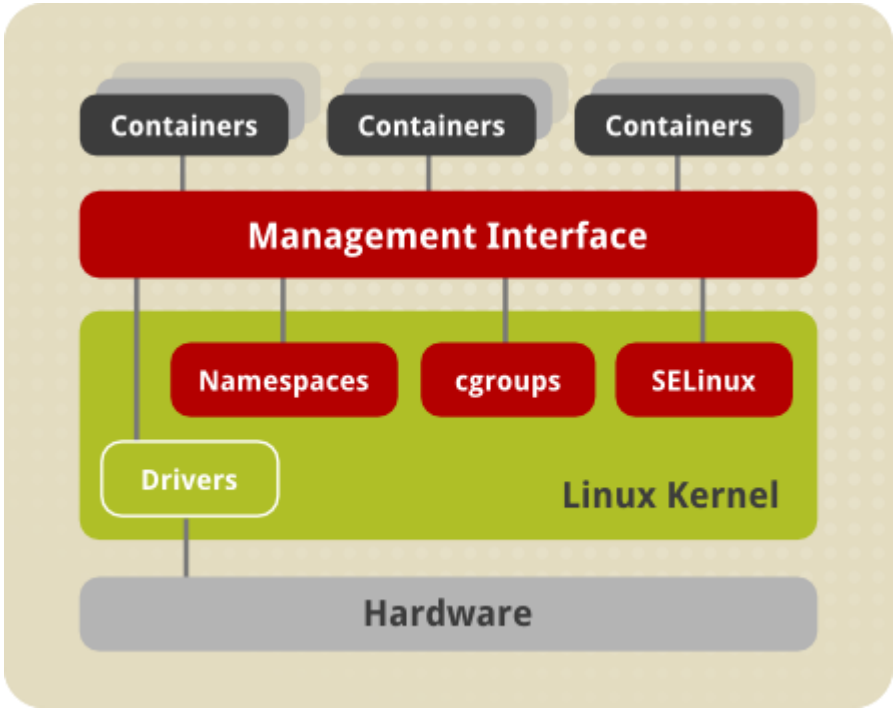
IT Agility

Rapid time-to-deployment

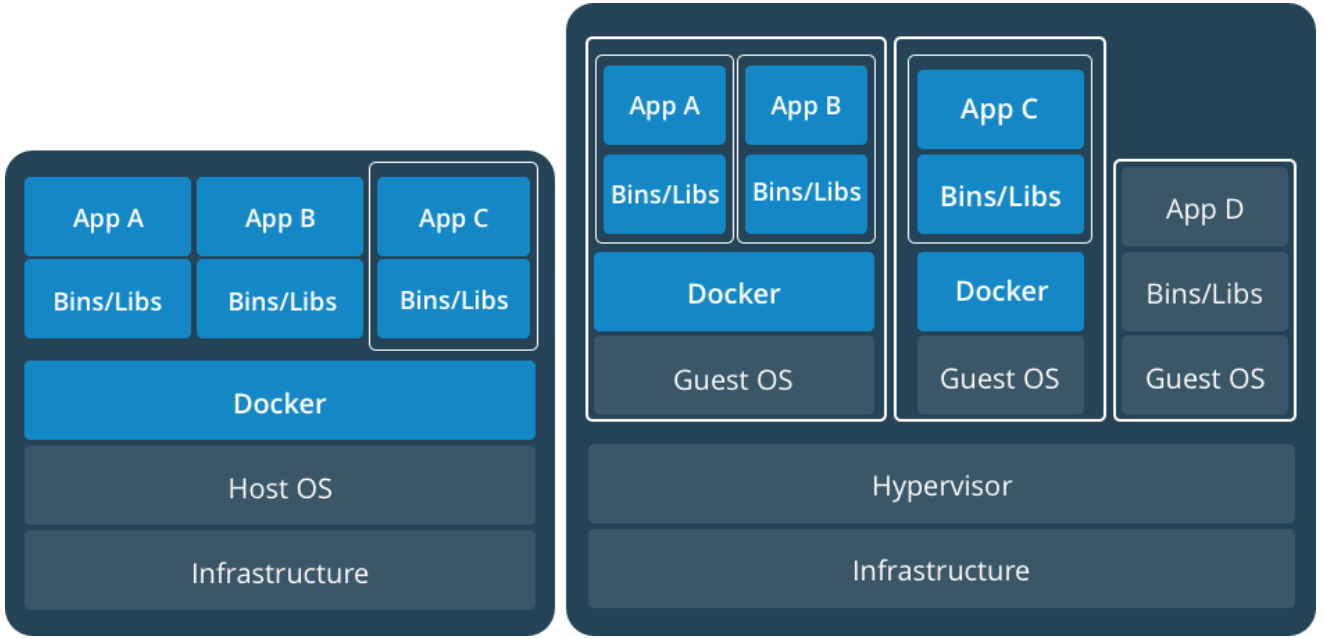
Ease of management

Continuous and dynamic

Containers



Linux Containers (LXC)



Containers Vs VMs

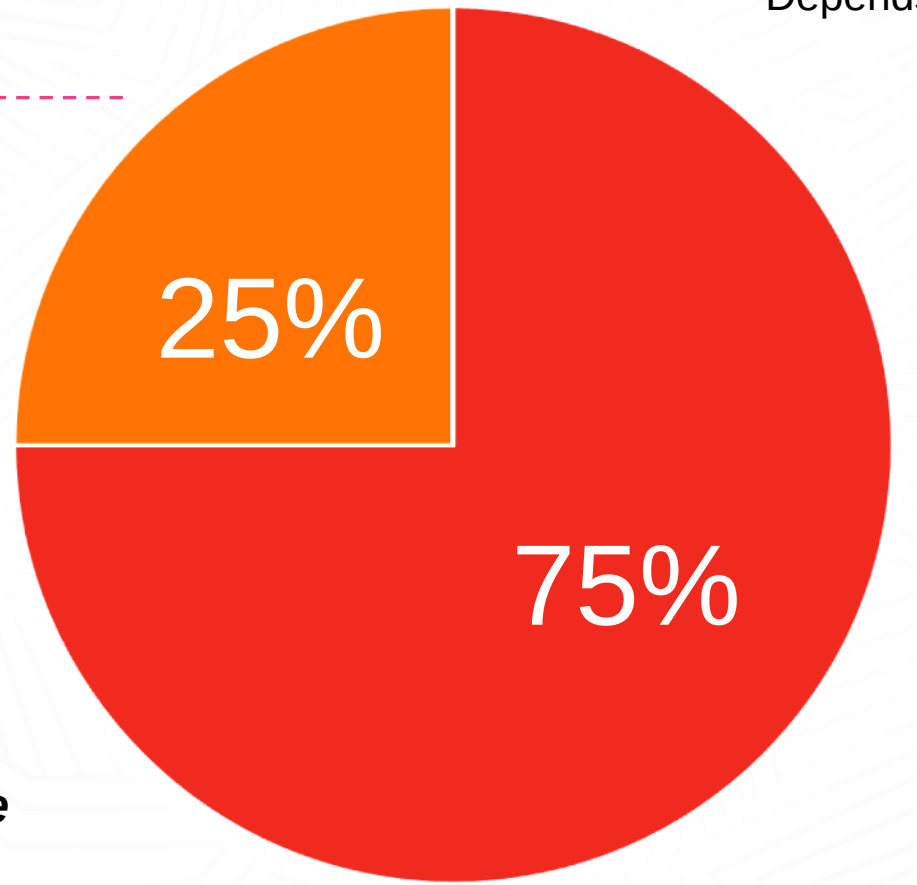
https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux_atomic_host/7/html/overview_of_containers_in_red_hat_systems/introduction_to_linux_containers
https://www.docker.com/what-container

Run Your Workloads, Where Its Best For You

- Some are best in public cloud, some on Premise
- Depends on need – amounts, time, costs

Elastic Workloads

Spin up and down resources on the public cloud



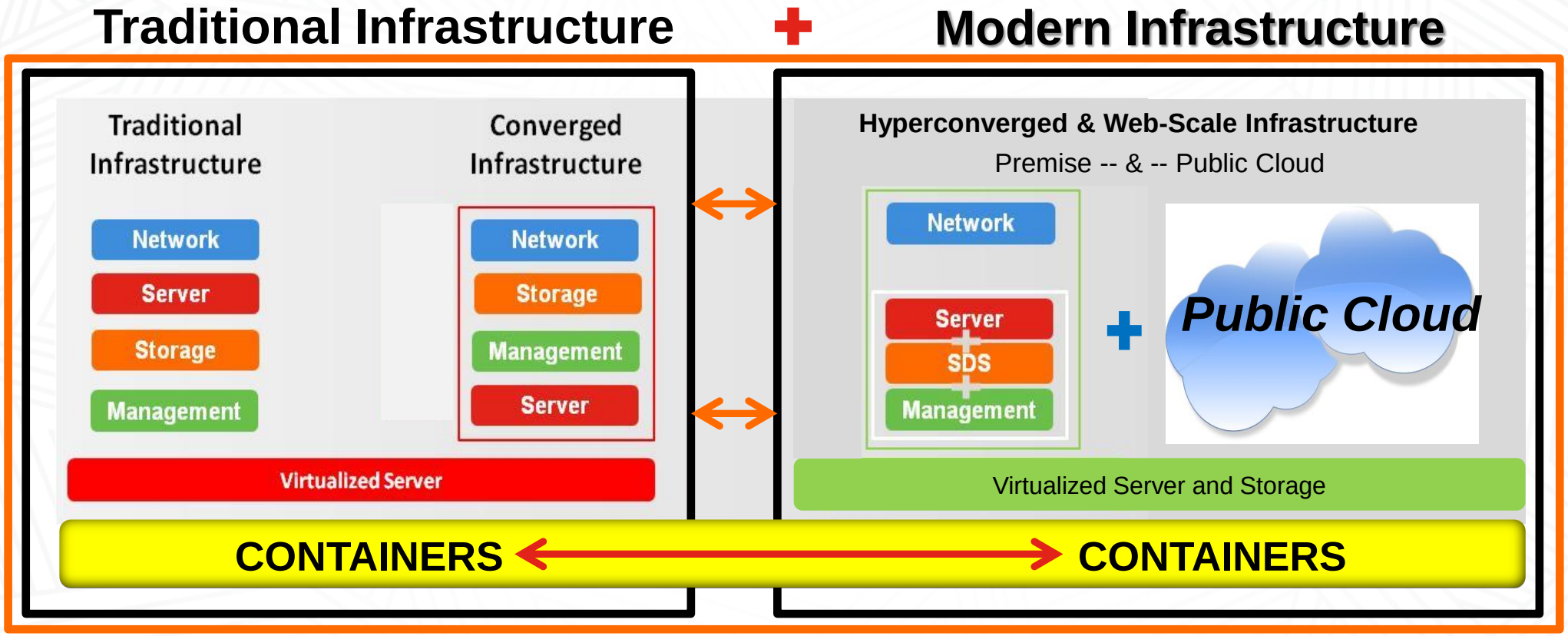
Predictable Workloads

Lower costs with private cloud infrastructure

Your % Will Vary – You Choose

Balance Owning and Renting For Today's Enterprise Workloads

Hybrid-Cloud – Integrating 3-Tier, Web-Scale, Public Cloud



- Maximize legacy infrastructure where cost-effective
- Enable Hybrid-Cloud infrastructure & cloud-native tools

- Leverage "Prem" and "Public" resources when needed for specific applications and workloads

Container Platform Architecture

Traditional, Stateful, and Microservices-based Apps

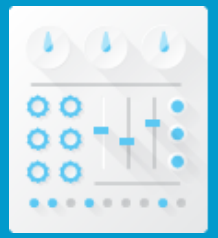
Business Automation	Integration	Data & Storage	Web & Mobile
Container	Container	Container	Container

Self-Service

Service Catalog
(Language Runtimes, Middleware, Databases)

Build Automation | Deployment Automation

OpenShift Application Lifecycle Management
(CI/CD)



Container Orchestration & Cluster Management
(kubernetes)

Networking | Storage | Registry | Logs & Metrics | Security

Infrastructure Automation & Cockpit

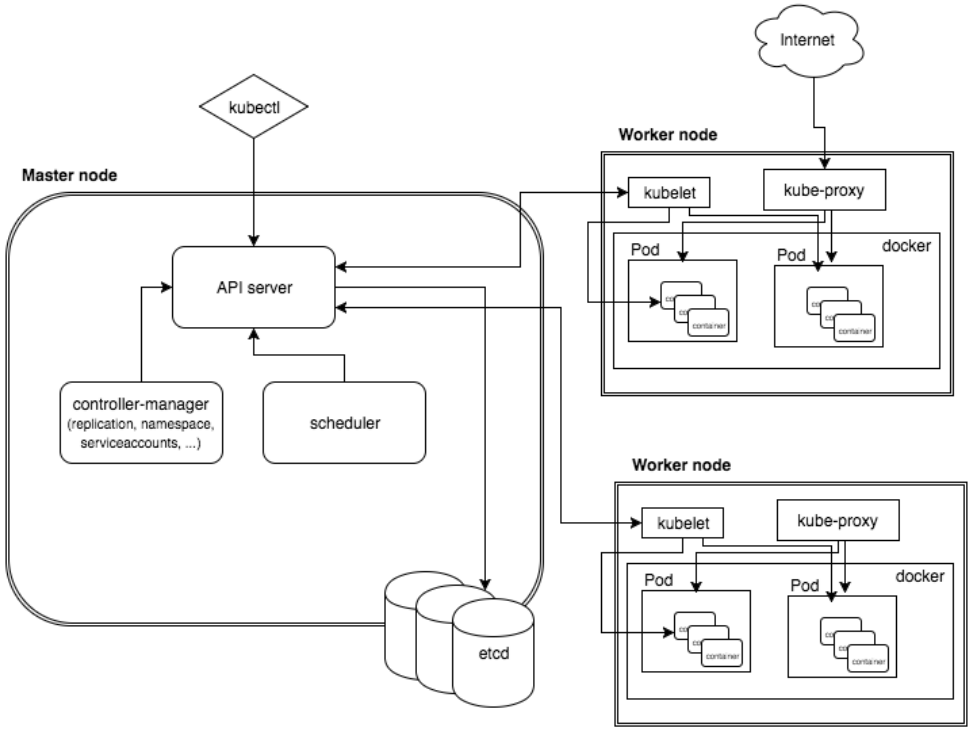
Enterprise Container Host

Container Runtime & Packaging
(Docker)

Atomic Host | Red Hat Enterprise Linux

redhat

Google Kubernetes Container Orchestrator



https://kubernetes.io/

Kubernetes Features

Automatic binpacking

Automatically places containers based on their resource requirements and other constraints, while not sacrificing availability. Mix critical and best-effort workloads in order to drive up utilization and save even more resources.

Self-healing

Restarts containers that fail, replaces and reschedules containers when nodes die, kills containers that don't respond to your user-defined health check, and doesn't advertise them to clients until they are ready to serve.

Horizontal scaling

Scale your application up and down with a simple command, with a UI, or automatically based on CPU usage.

Service discovery and load balancing

No need to modify your application to use an unfamiliar service discovery mechanism. Kubernetes gives containers their own IP addresses and a single DNS name for a set of containers, and can load-balance across them.

Automated rollouts and rollbacks

Kubernetes progressively rolls out changes to your application or its configuration, while monitoring application health to ensure it doesn't kill all your instances at the same time. If something goes wrong, Kubernetes will rollback the change for you. Take advantage of a growing ecosystem of deployment solutions.

Secret and configuration management

Deploy and update secrets and application configuration without rebuilding your image and without exposing secrets in your stack configuration.

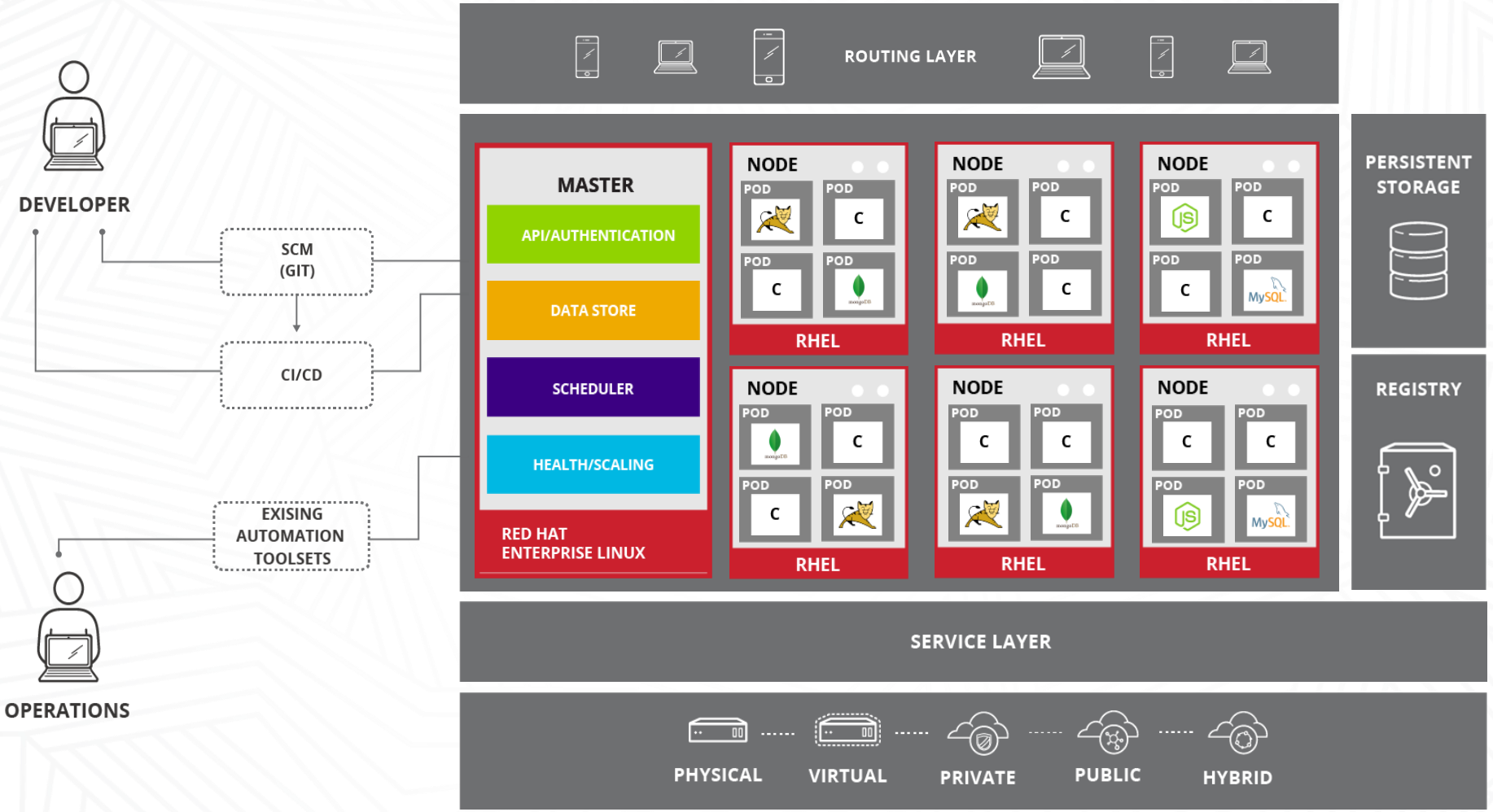
Storage orchestration

Automatically mount the storage system of your choice, whether from local storage, a public cloud provider such as GCP or AWS, or a network storage system such as NFS, iSCSI, Gluster, Ceph, Cinder, or Flocker.

Batch execution

In addition to services, Kubernetes can manage your batch and CI workloads, replacing containers that fail, if desired.

OpenShift Platform Architecture



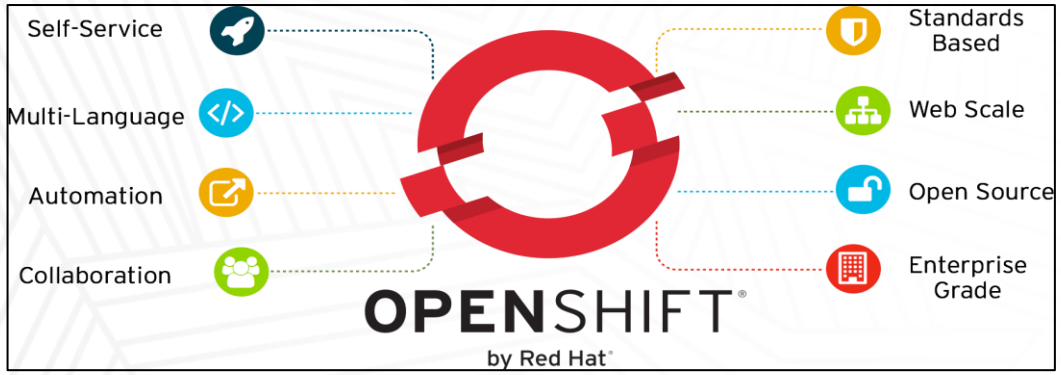
https://docs.openshift.com/container-platform/3.5/architecture/index.html

Implementing OpenShift Container Platform

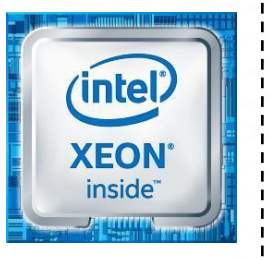
Lenovo Container Orchestration Platform

An integrated HW and SW reference architecture, purpose built for container applications. Easy to deploy, easy to scale, easy to manage.

- *Integrated high speed "IP fabric", capable of delivering fast and predictable performance.*
- Integrated container application platform (PaaS) reference architecture with enterprise grade OpenShift/Kubernetes and xClarity infrastructure management
- Simplified deployment (minutes), easily scale across racks
- Highly optimized for scalability, performance and value (Intel SSD technology, Intel Xeon, Intel NIC's)



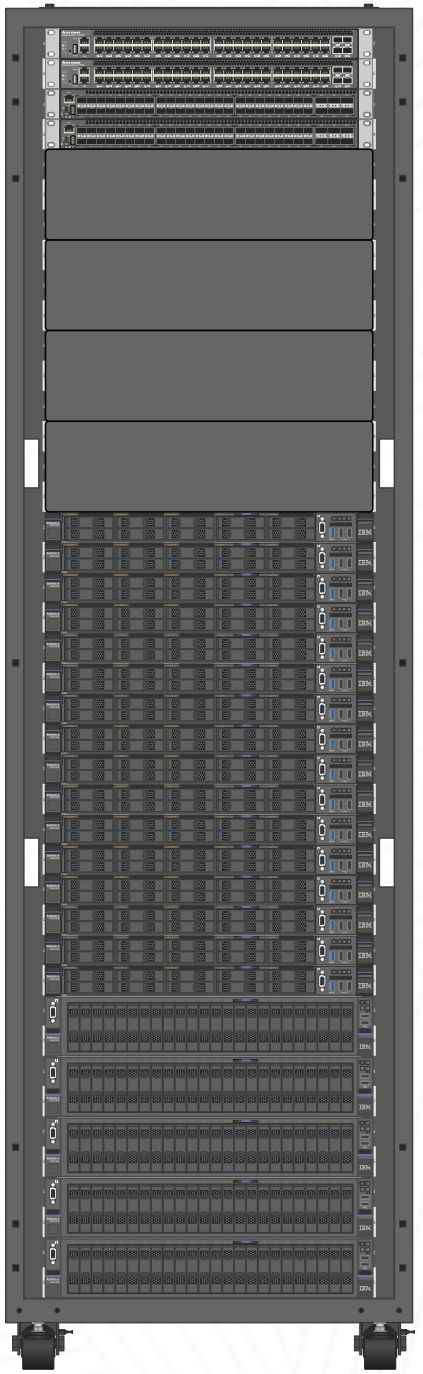
redhat



CPUs

SSDs

Network



Leaf/spine switch architecture to scale racks horizontally

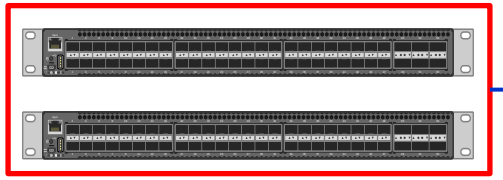
Seamless scaling of compute or storage capacity as you grow

Lenovo x3550 M5 Server for compute building block

Lenovo x3650 M5 Server for SDS storage building block

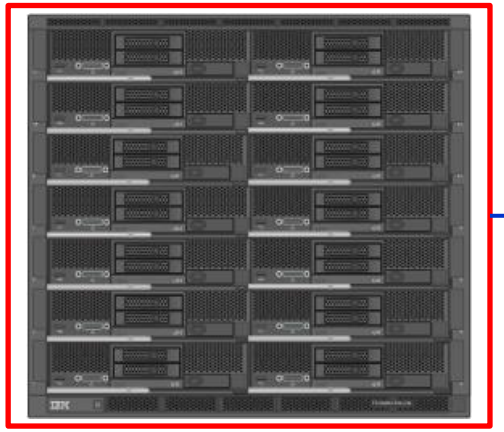
Hardware Architecture and Considerations for OpenShift

Network Leaf/Spine Architecture



HA Management Cluster

Scalable Compute Cluster



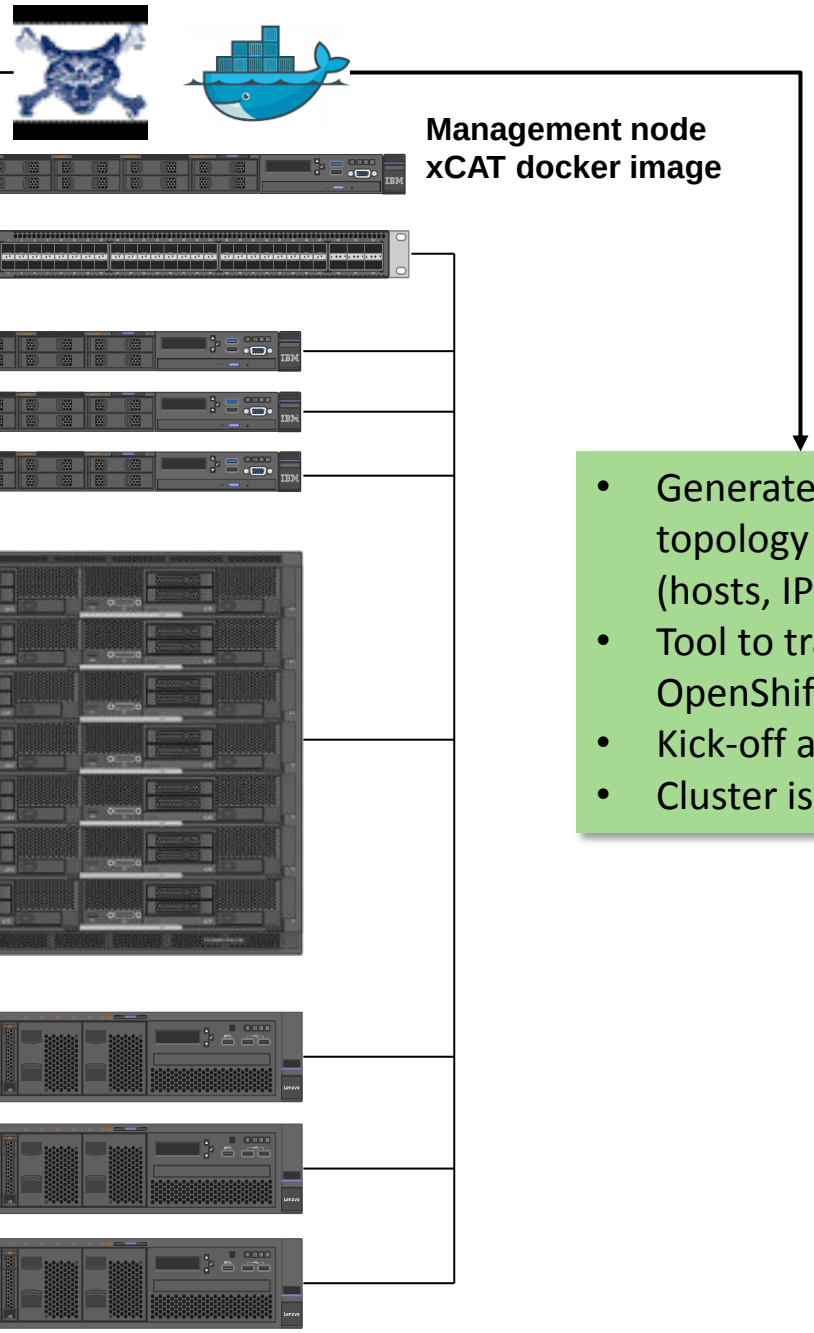
Scalable Storage Cluster

Role	Style	Notes
etcd	Active-active	Fully redundant deployment with load balancing
API Server	Active-active	Managed by HAProxy
Controller Manager Server	Active-passive	One instance is elected as a cluster leader at a time
HAProxy	Active-passive	Balances load between API master endpoints

A node provides the runtime environments for containers. Each node in a Kubernetes cluster has the required services to be managed by the master. Nodes also have the required services to run pods, including the Docker service, a kubelet, and a service proxy.

OpenShift Container Platform leverages the Kubernetes persistent volume (PV) framework to allow administrators to provision persistent storage for a cluster. Using persistent volume claims (PVCs), developers can request PV resources without having specific knowledge of the underlying storage infrastructure.

OpenShift Deployment Automation



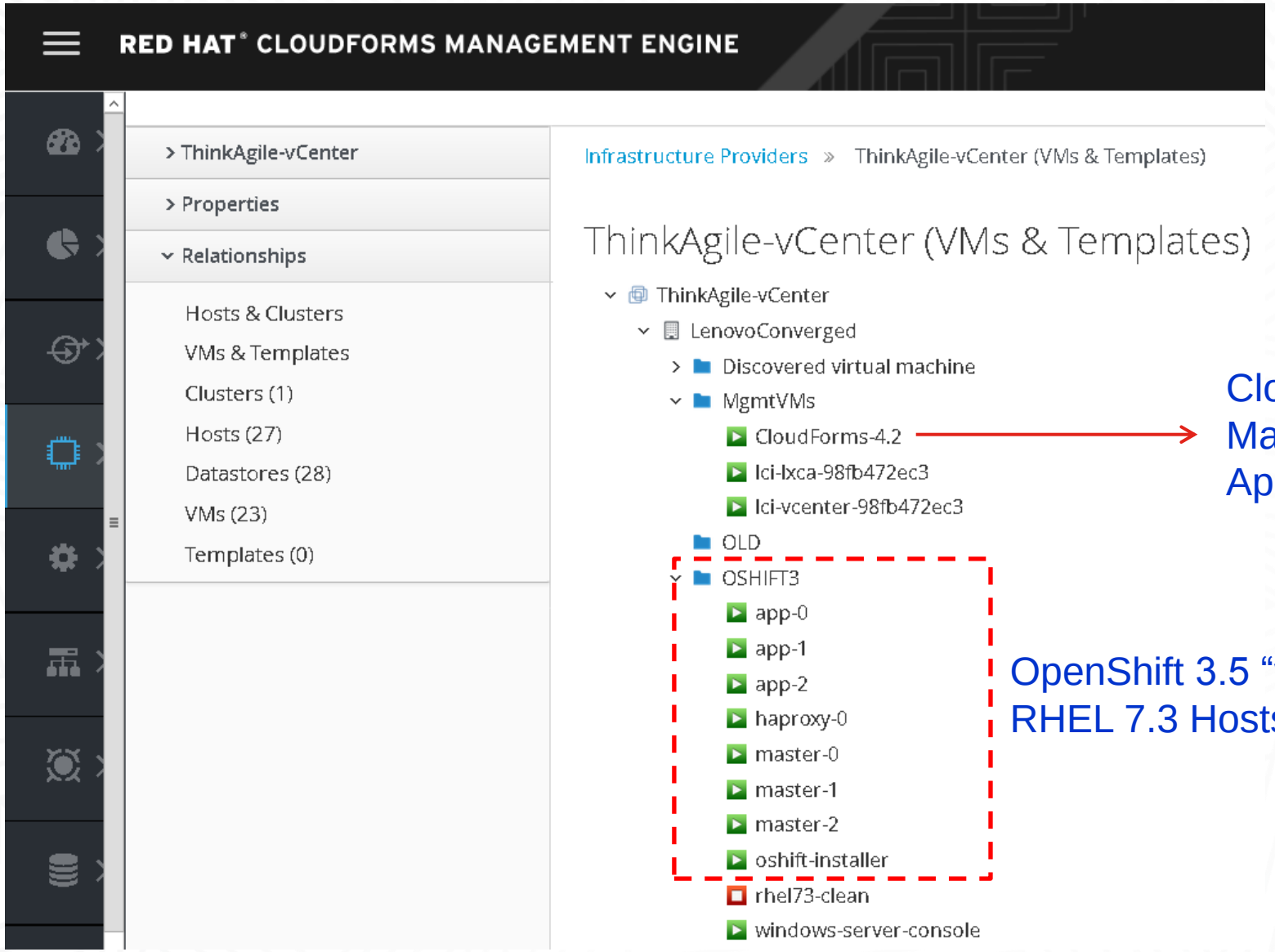
- Baremetal initial discovery
- IMM/IPMI configuration
- Firmware updates
- Network configuration
- OS deployment
- Embedded DNS, DHCP,TFTP

- Generates cluster HW manifest and topology
  (hosts, IP addresses, inventory, etc.)
- Tool to translate topology to Ansible OpenShift configuration file
- Kick-off ansible installation playbook
- Cluster is deployed and ready for operation

/etc/ansible/hosts

```
# Create an OSEv3 group that contains the master, nodes, etcd, and lb groups.
# The lb group lets Ansible configure HAProxy as the load balancing solution.
# Comment lb out if your load balancer is pre-configured.
[OSEv3:children]
masters
nodes
etcd
lb

# Set variables common for all OSEv3 hosts
[OSEv3:vars]
ansible_ssh_user=root
deployment_type=openshift-enterprise
openshift_master_default_subdomain=apps.oshift.local

openshift_master_identity_providers=[{'name': 'htpasswd_auth', 'login': 'true', 'challenge': 'true', 'kind': 'HTPasswdPasswordIdentityProvider',
'filename': '/etc/origin/master/htpasswd'}]

openshift_master_cluster_method=native
openshift_master_cluster_hostname=haproxy-0
openshift_master_cluster_public_hostname=haproxy-0.oshift.local
openshift_hosted_metrics_public_url=master-0.oshift.local

# enable ntp on masters to ensure proper failover
openshift_clock_enabled=true

# host group for masters
[masters]
master-0
master-1
master-2

# host group for etcd
[etcd]
master-0
master-1
master-2

# Specify load balancer host
[lb]
haproxy-0

# host group for nodes, includes region info
[nodes]
master-[0:2] openshift_node_labels="{'region': 'infra', 'zone': 'default'}"
app-0 openshift_node_labels="{'region': 'primary', 'zone': 'east'}"
app-1 openshift_node_labels="{'region': 'primary', 'zone': 'west'}"
app-2 openshift_node_labels="{'region': 'primary', 'zone': 'west'}"
```

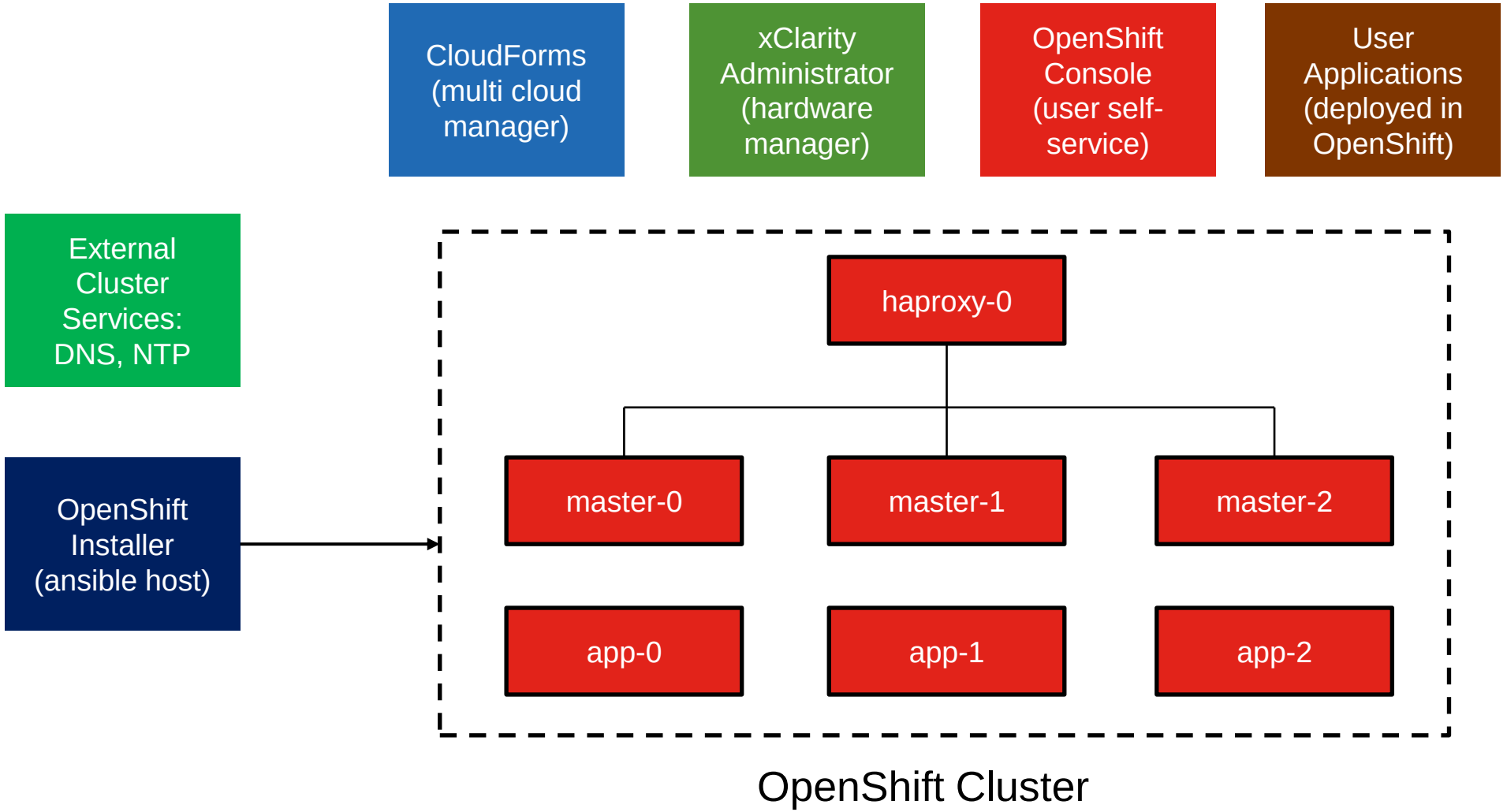
Virtualized OpenShift Enterprise 3.5 Implementation



CloudForms Management Engine Appliance

OpenShift 3.5 “virtual” Cluster RHEL 7.3 Hosts

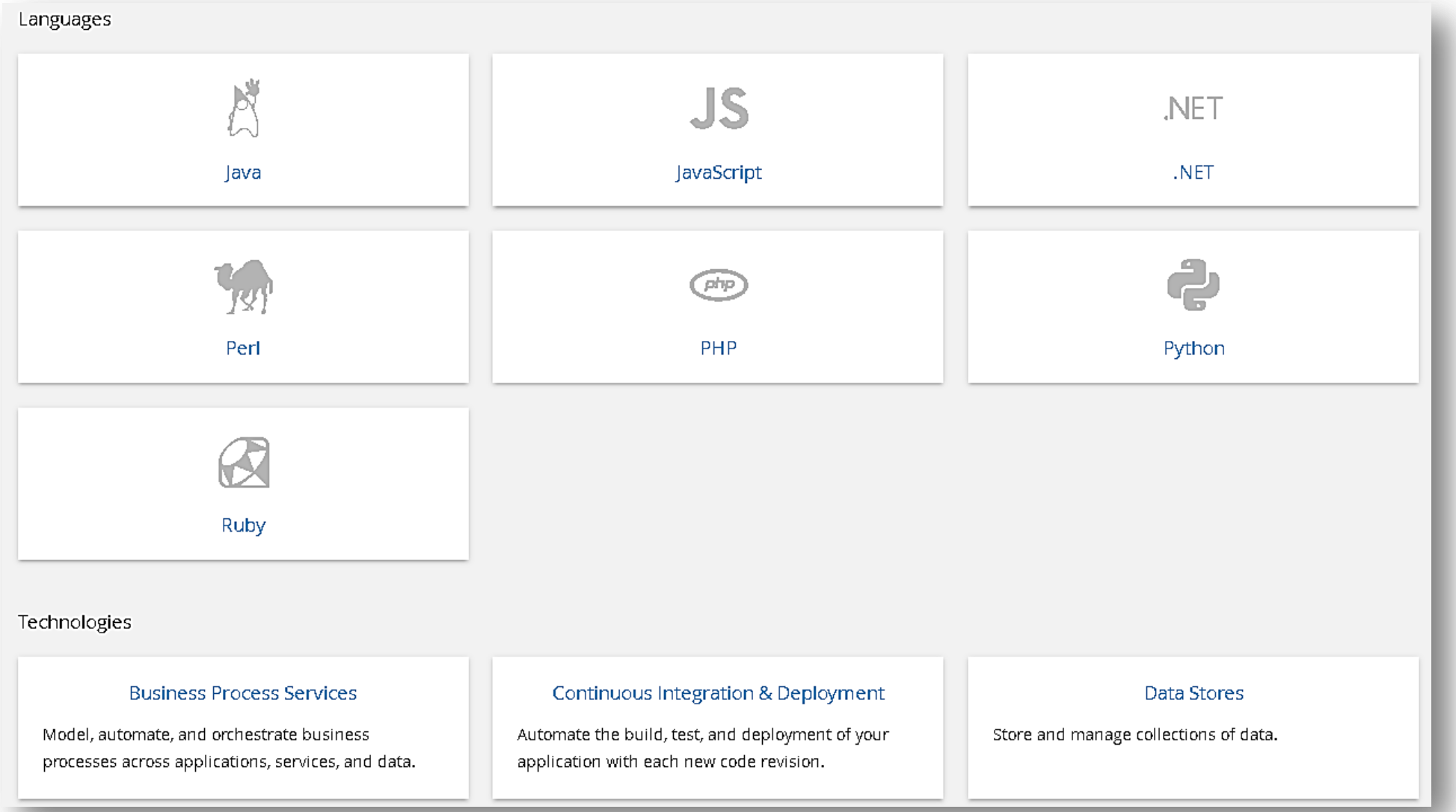
OpenShift Deployment Architecture

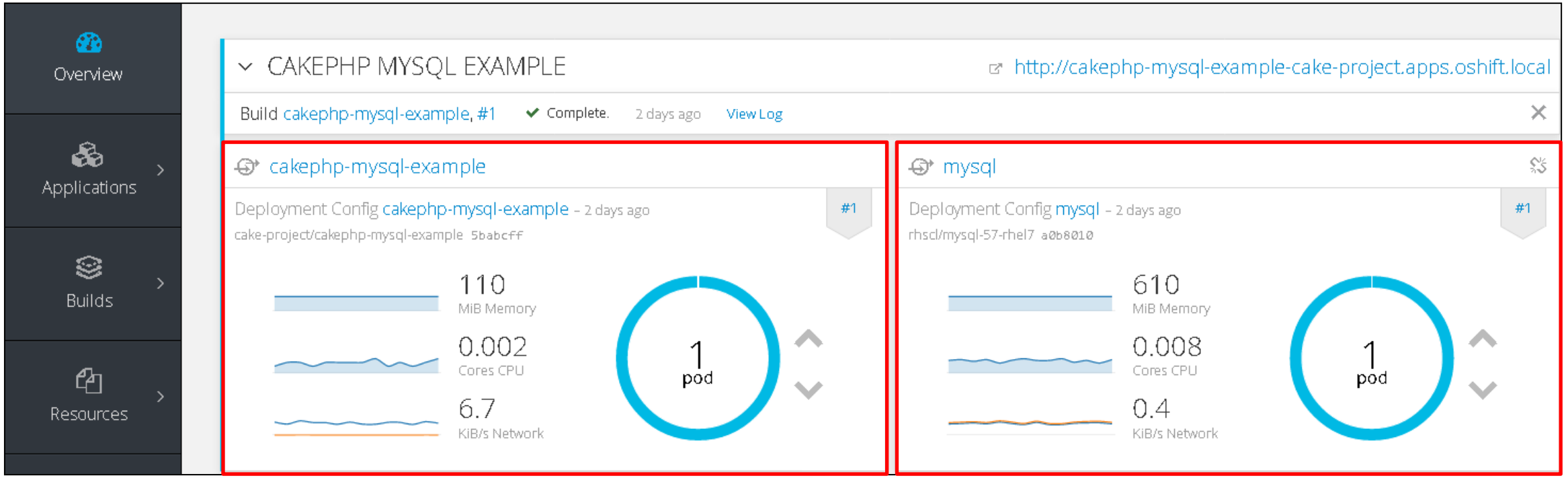


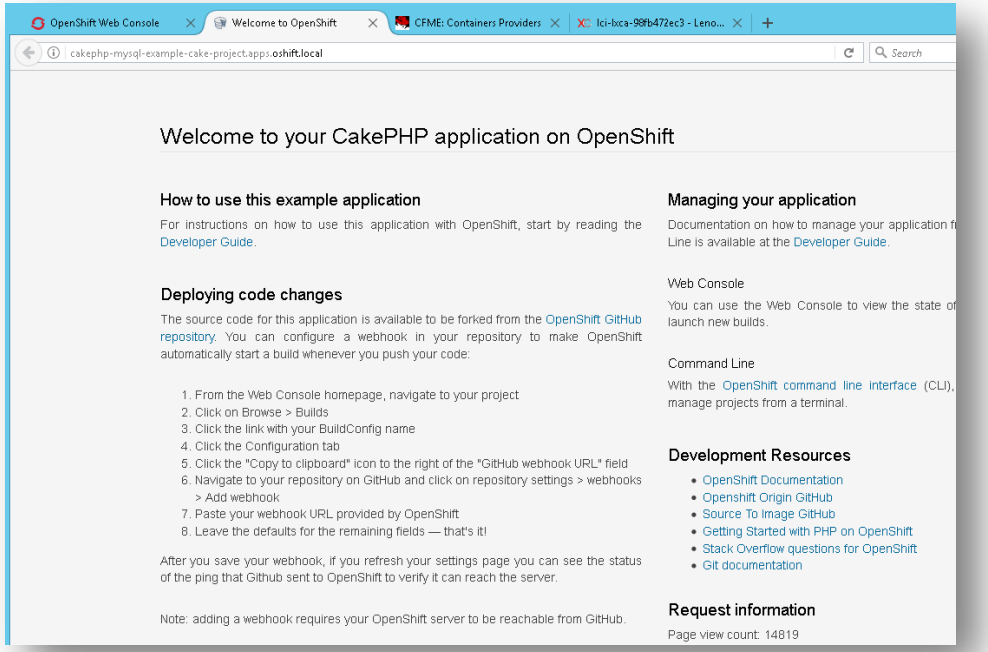
```
[root@master-0 ~]# oc get nodes --show-labels
NAME                 STATUS    AGE       LABELS
app-0.oshift.local   Ready     1d        beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,kubernetes.io/hostname=app-0.oshift.local,region=primary,zone=east
app-1.oshift.local   Ready     1d        beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,kubernetes.io/hostname=app-1.oshift.local,region=primary,zone=west
app-2.oshift.local   Ready     1d        beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,kubernetes.io/hostname=app-2.oshift.local,region=primary,zone=west
master-0.oshift.local   Ready     1d        beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,kubernetes.io/hostname=master-0.oshift.local,region=infra,zone=default
master-1.oshift.local   Ready     1d        beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,kubernetes.io/hostname=master-1.oshift.local,region=infra,zone=default
master-2.oshift.local   Ready     1d        beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,kubernetes.io/hostname=master-2.oshift.local,region=infra,zone=default

[root@master-0 ~]# oc get projects
NAME               DISPLAY NAME     STATUS
cake-project       cake-phy-mysql   Active
default                             Active
kube-system                         Active
logging                             Active
management-infra                    Active
nexus3-demo        nexus3-example   Active
openshift                           Active
openshift-infra                     Active

[root@master-0 ~]# oc get pods --all-namespaces
NAMESPACE          NAME                            READY     STATUS      RESTARTS   AGE
cake-project       cakephp-mysql-example-1-build   0/1       Completed   0          1d
cake-project       cakephp-mysql-example-1-d6zcr   1/1       Running     0          1d
cake-project       mysql-1-8tmpf                   1/1       Running     0          1d
default            docker-registry-1-350h2         1/1       Running     0          1d
default            registry-console-1-2lwbt        1/1       Running     0          1d
default            router-1-8xxkv                  1/1       Running     0          1d
default            router-1-g45qs                  1/1       Running     0          1d
default            router-1-vmh3s                  1/1       Running     0          1d
nexus3-demo        nexus3-1-vcs0p                  1/1       Running     0          1d
nexus3-demo        nexus3-2-deploy                 0/1       Error       0          1d
openshift          cakephp-example-1-build         0/1       Completed   0          1d
openshift          cakephp-example-1-wttb5         1/1       Running     0          1d
openshift-infra    hawkular-cassandra-1-11gt4      1/1       Running     0          1d
openshift-infra    hawkular-metrics-p5z89          1/1       Running     3          1d
openshift-infra    heapster-blmsw                  1/1       Running     0          1d


[root@master-0 ~]# oc get routes --all-namespaces
NAMESPACE          NAME                  HOST/PORT                                          PATH      SERVICES              PORT                    TERMINATION   WILDCARD
cake-project       cakephp-mysql-example   cakephp-mysql-example-cake-project.apps.oshift.local             cakephp-mysql-example   <all>                                 None
default            docker-registry         docker-registry-default.apps.oshift.local                        docker-registry         <all>                   passthrough   None
default            registry-console        registry-console-default.apps.oshift.local                       registry-console        <all>                   passthrough   None
nexus3-demo        nexus3-route            nx3.apps.oshift.local                                            nexus3                  8081-tcp                              None
openshift          cakephp-example         cakephp-example-openshift.apps.oshift.local                      cakephp-example         <all>                                 None
openshift-infra    hawkular-metrics        master-0.oshift.local                                            hawkular-metrics        https-end        point   reencrypt     None
```

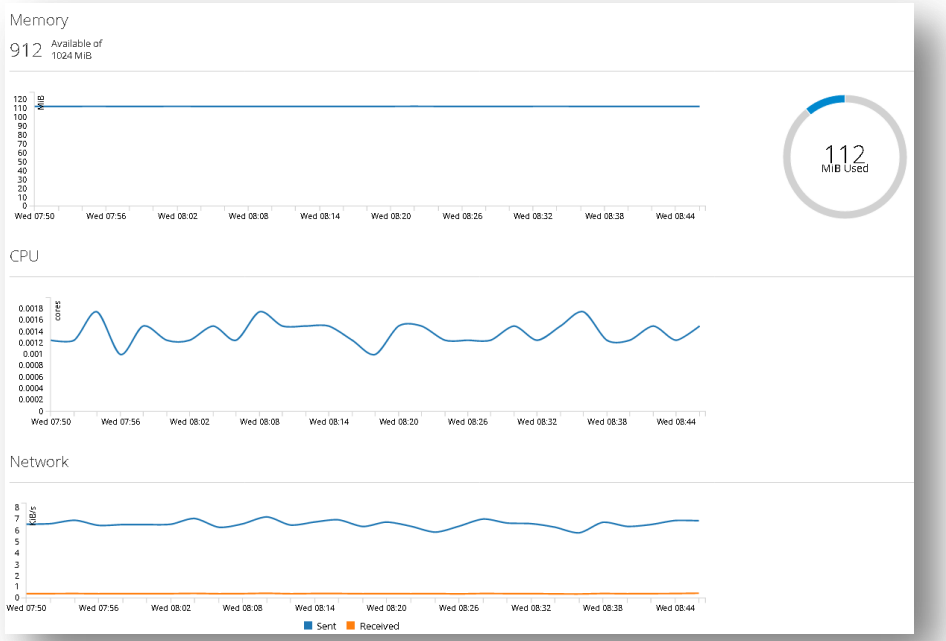
Creating and deploying your containerized apps

Built-in templates for common runtimes and usecases

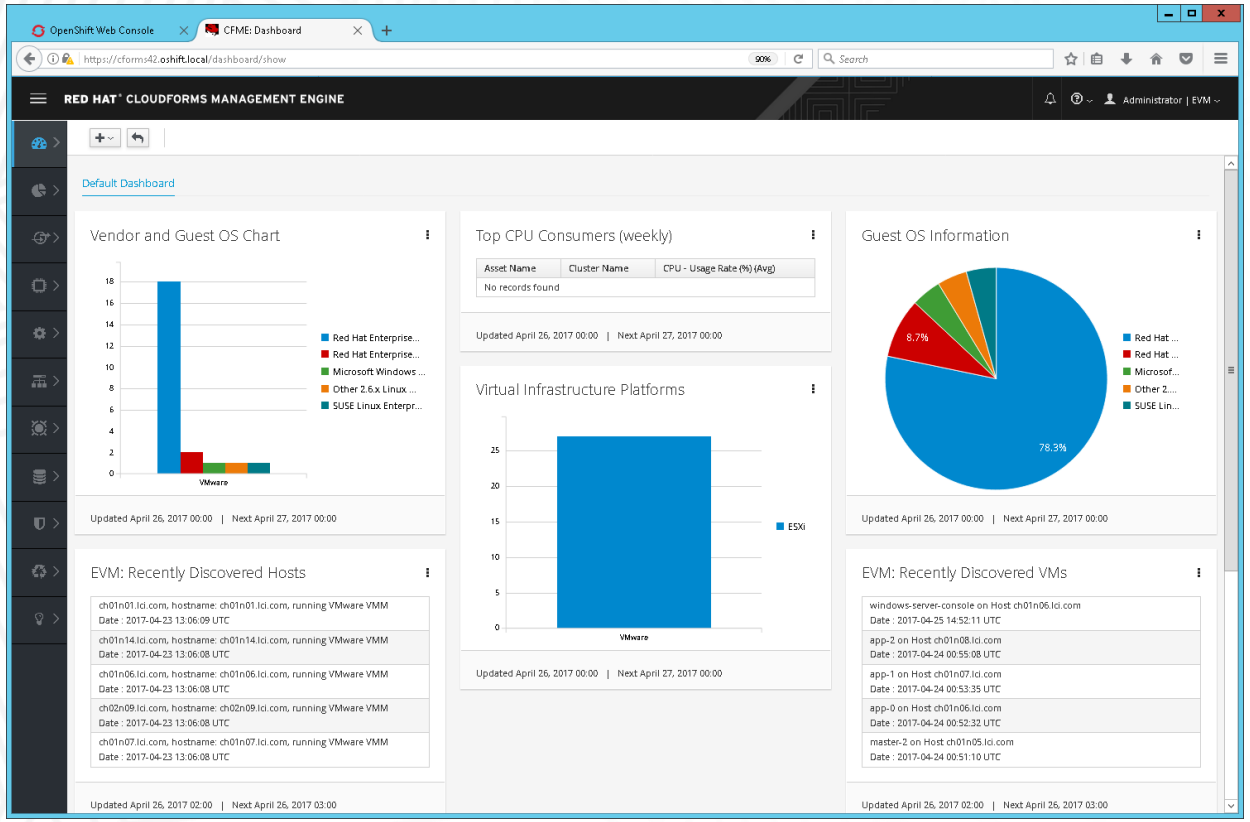


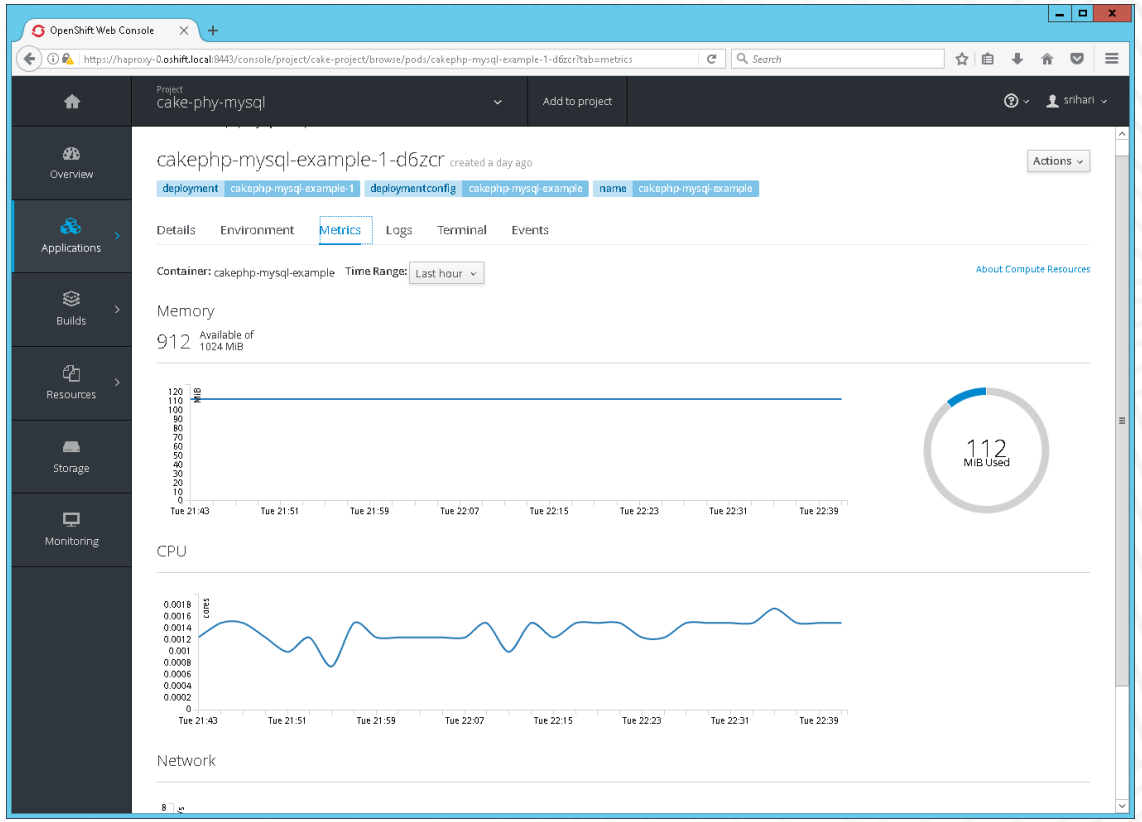

Overview
Applications
Builds
Resources
CAKEPHP MYSQL EXAMPLE
http://cakephp-mysql-example-cake-project.apps.oshift.local
Build cakephp-mysql-example, #1
Complete.
2 days ago
View Log
cakephp-mysql-example
Deployment Config cakephp-mysql-example – 2 days ago
cake-project/cakephp-mysql-example 5babcff
#1
110
MiB Memory
0.002
Cores CPU
6.7
KiB/s Network
1
pod
mysql
Deployment Config mysql – 2 days ago
rhscl/mysql-57-rhel7 a0b8010
#1
610
MiB Memory
0.008
Cores CPU
0.4
KiB/s Network
1
pod


OpenShift Web Console
Welcome to OpenShift
CFME: Containers Providers
cakephp-mysql-example-cake-project.apps.oshift.local
Welcome to your CakePHP application on OpenShift
How to use this example application
For instructions on how to use this application with OpenShift, start by reading the Developer Guide.
Deploying code changes
The source code for this application is available to be forked from the OpenShift GitHub repository. You can configure a webhook in your repository to make OpenShift automatically start a build whenever you push your code:
1. From the Web Console homepage, navigate to your project
2. Click on Browse > Builds
3. Click the link with your BuildConfig name
4. Click the Configuration tab
5. Click the "Copy to clipboard" icon to the right of the "GitHub webhook URL" field
6. Navigate to your repository on GitHub and click on repository settings > webhooks > Add webhook
7. Paste your webhook URL provided by OpenShift
8. Leave the defaults for the remaining fields — that's it!
After you save your webhook, if you refresh your settings page you can see the status of the ping that Github sent to OpenShift to verify it can reach the server.
Note: adding a webhook requires your OpenShift server to be reachable from GitHub.
Managing your application
Documentation on how to manage your application f
Line is available at the Developer Guide.
Web Console
You can use the Web Console to view the state of
launch new builds.
Command Line
With the OpenShift command line interface (CLI),
manage projects from a terminal.
Development Resources
OpenShift Documentation
Openshift Origin GitHub
Source To Image GitHub
Getting Started with PHP on OpenShift
Stack Overflow questions for OpenShift
Git documentation
Request information
Page view count: 14819

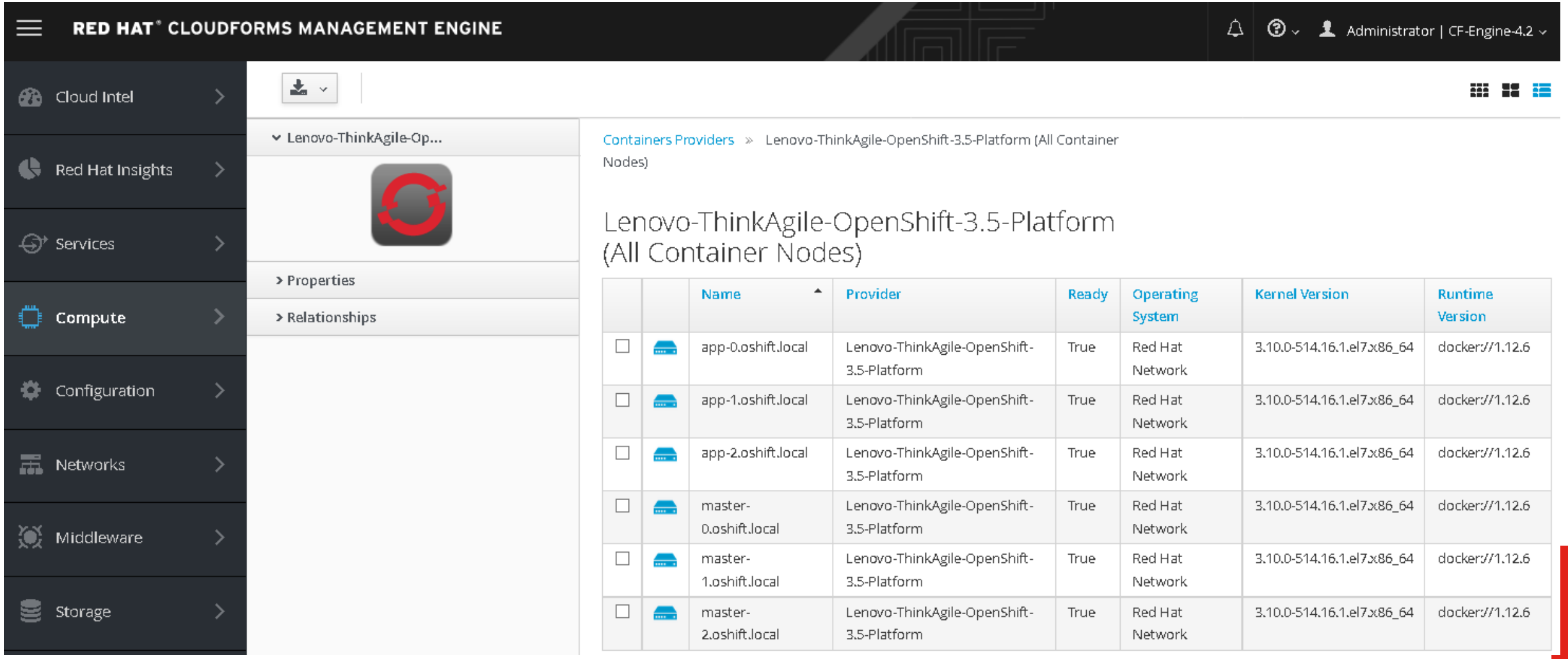

Memory
912 Available of 1024 MiB
112
MiB Used
CPU
Network
Sent
Received

Managing your Hybrid Cloud with CloudForms

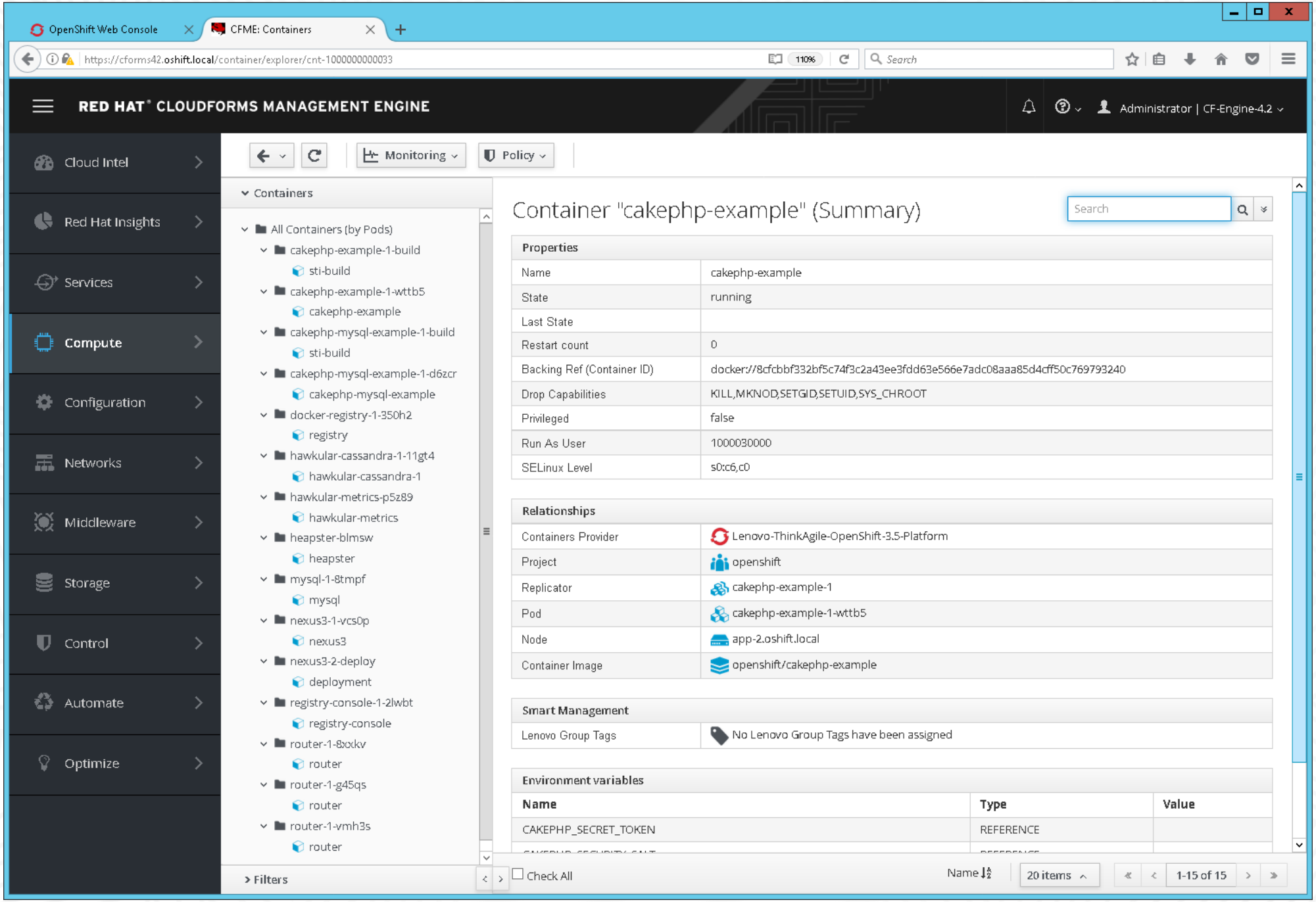




Managing OpenShift with CloudForms



Name	Provider	Ready	Operating System	Kernel Version	Runtime Version
app-0.oshift.local	Lenovo-ThinkAgile-OpenShift-3.5-Platform	True	Red Hat Network	3.10.0-514.16.1.el7x86_64	docker://1.12.6
app-1.oshift.local	Lenovo-ThinkAgile-OpenShift-3.5-Platform	True	Red Hat Network	3.10.0-514.16.1.el7x86_64	docker://1.12.6
app-2.oshift.local	Lenovo-ThinkAgile-OpenShift-3.5-Platform	True	Red Hat Network	3.10.0-514.16.1.el7x86_64	docker://1.12.6
master-0.oshift.local	Lenovo-ThinkAgile-OpenShift-3.5-Platform	True	Red Hat Network	3.10.0-514.16.1.el7x86_64	docker://1.12.6
master-1.oshift.local	Lenovo-ThinkAgile-OpenShift-3.5-Platform	True	Red Hat Network	3.10.0-514.16.1.el7x86_64	docker://1.12.6
master-2.oshift.local	Lenovo-ThinkAgile-OpenShift-3.5-Platform	True	Red Hat Network	3.10.0-514.16.1.el7x86_64	docker://1.12.6


OpenShift Web Console
CFME: Containers
https://cforms42.oshift.local/container/explorer/cnt-1000000000033
RED HAT® CLOUDFORMS MANAGEMENT ENGINE
Administrator | CF-Engine-4.2
Cloud Intel
Red Hat Insights
Services
Compute
Configuration
Networks
Middleware
Storage
Control
Automate
Optimize
Monitoring
Policy
Containers
All Containers (by Pods)
cakephp-example-1-build
sti-build
cakephp-example-1-wttb5
cakephp-example
cakephp-mysql-example-1-build
sti-build
cakephp-mysql-example-1-d6zcr
cakephp-mysql-example
docker-registry-1-350h2
registry
hawkular-cassandra-1-11gt4
hawkular-cassandra-1
hawkular-metrics-p5z89
hawkular-metrics
heapster-blmsw
heapster
mysql-1-8tmpf
mysql
nexus3-1-vcs0p
nexus3
nexus3-2-deploy
deployment
registry-console-1-2lwbt
registry-console
router-1-8xxkv
router
router-1-g45qs
router
router-1-vmh3s
router
Filters
Check All
Container "cakephp-example" (Summary)
Search
Properties
Name
cakephp-example
State
running
Last State
Restart count
0
Backing Ref (Container ID)
docker://8cfcbbf332bf5c74f3c2a43ee3fdd63e566e7adc08aaa85d4cff50c769793240
Drop Capabilities
KILL,MKNOD,SETGID,SETUID,SYS_CHROOT
Privileged
false
Run As User
1000030000
SELinux Level
s0:c6,c0
Relationships
Containers Provider
Lenovo-ThinkAgile-OpenShift-3.5-Platform
Project
openshift
Replicator
cakephp-example-1
Pod
cakephp-example-1-wttb5
Node
app-2.oshift.local
Container Image
openshift/cakephp-example
Smart Management
Lenovo Group Tags
No Lenovo Group Tags have been assigned
Environment variables
Name
Type
Value
CAKEPHP_SECRET_TOKEN
REFERENCE
Name
20 items
1-15 of 15

OpenShift metrics integration for chargeback

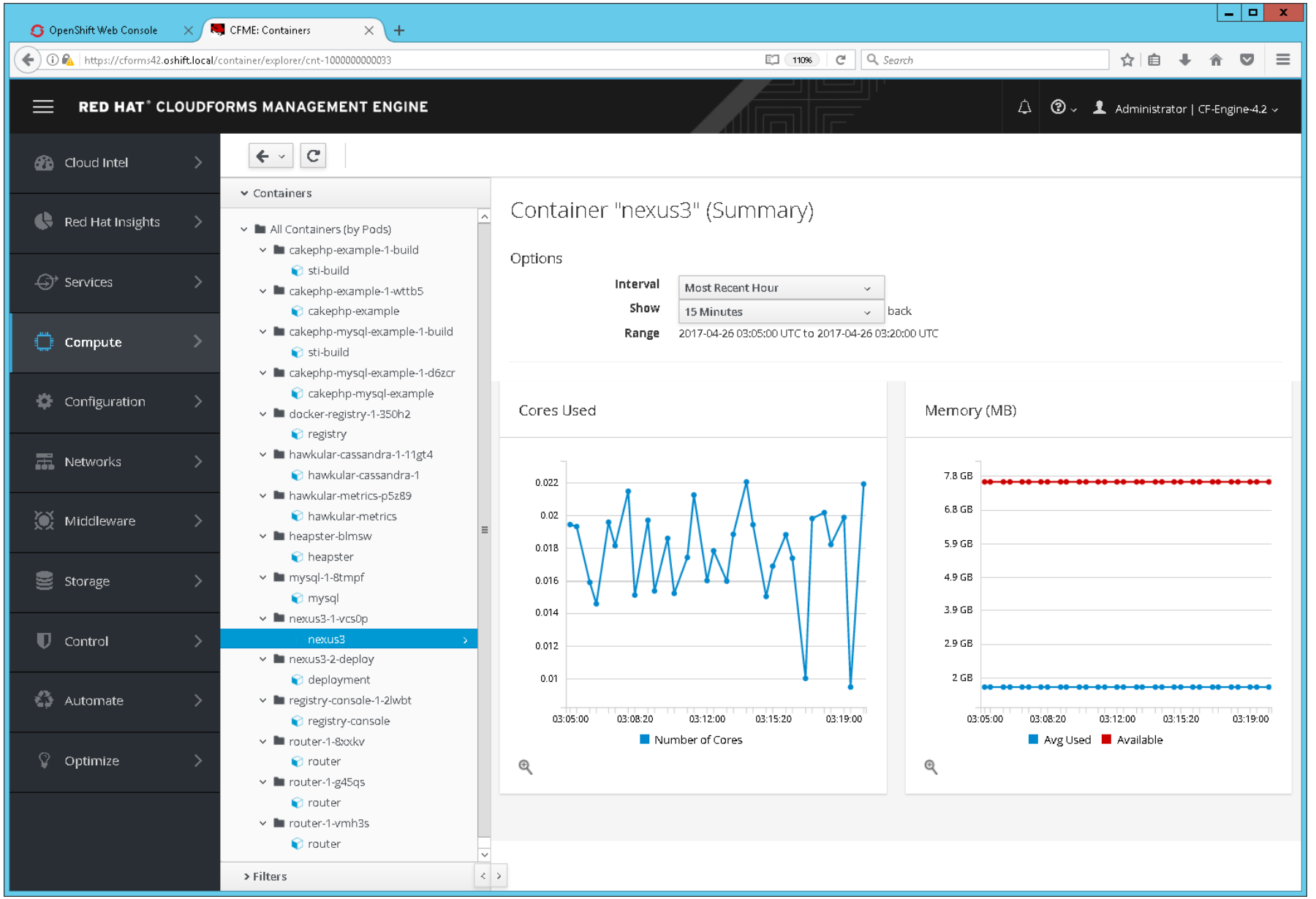
Hawkular Metrics

A time series metrics engine based on Cassandra

0.21.5.Final-redhat-1

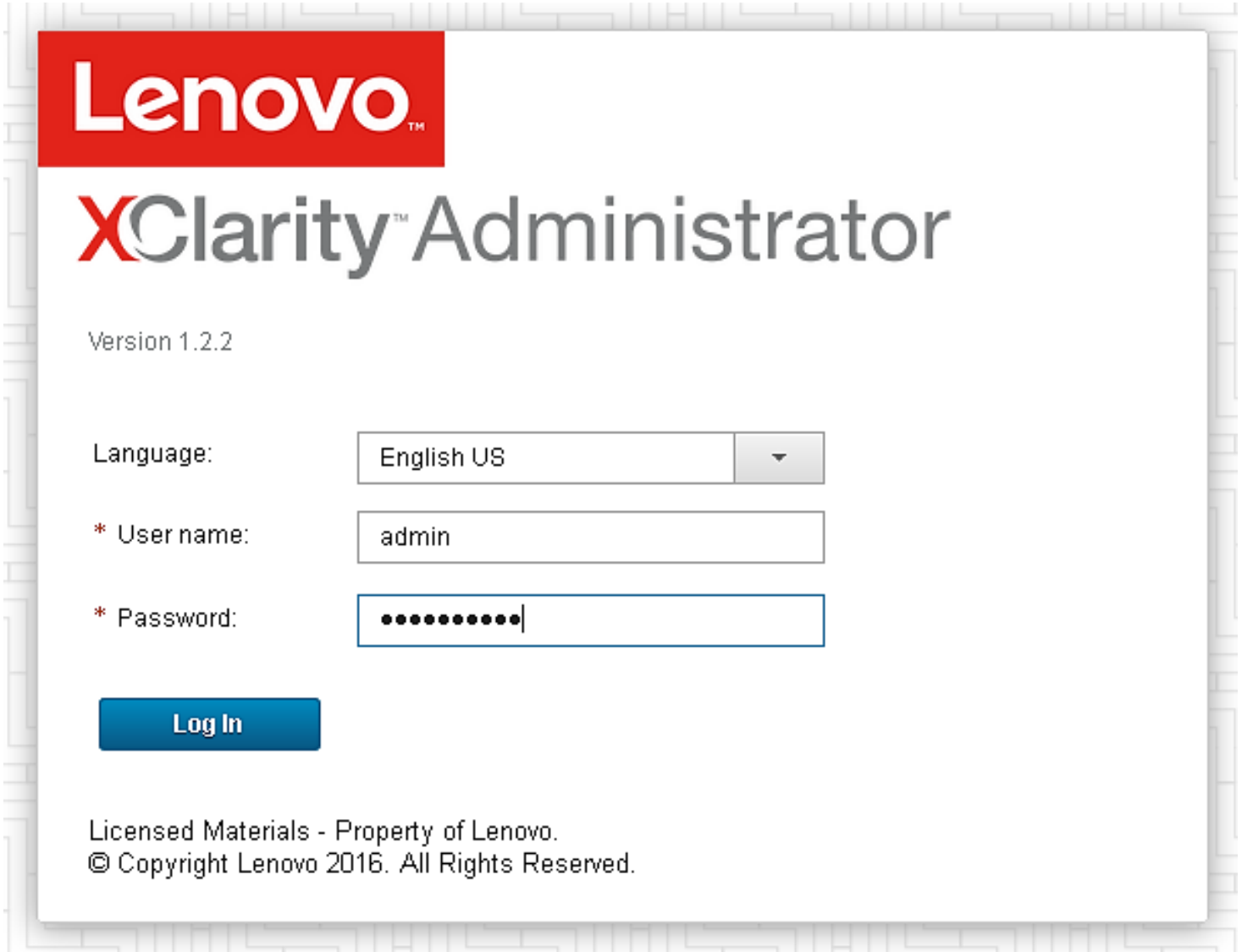
(Git SHA1 - 632f908a52d3e45b3a0bafa84e117ec6ca87bb19)

Metrics Service :STARTED



Lenovo Converged Infrastructure

Managing your infrastructure with Lenovo xClarity Administrator


Lenovo
XClarity Administrator
Version 1.2.2
Language:
English US
* User name:
admin
* Password:
Log In
Licensed Materials - Property of Lenovo.
© Copyright Lenovo 2016. All Rights Reserved.

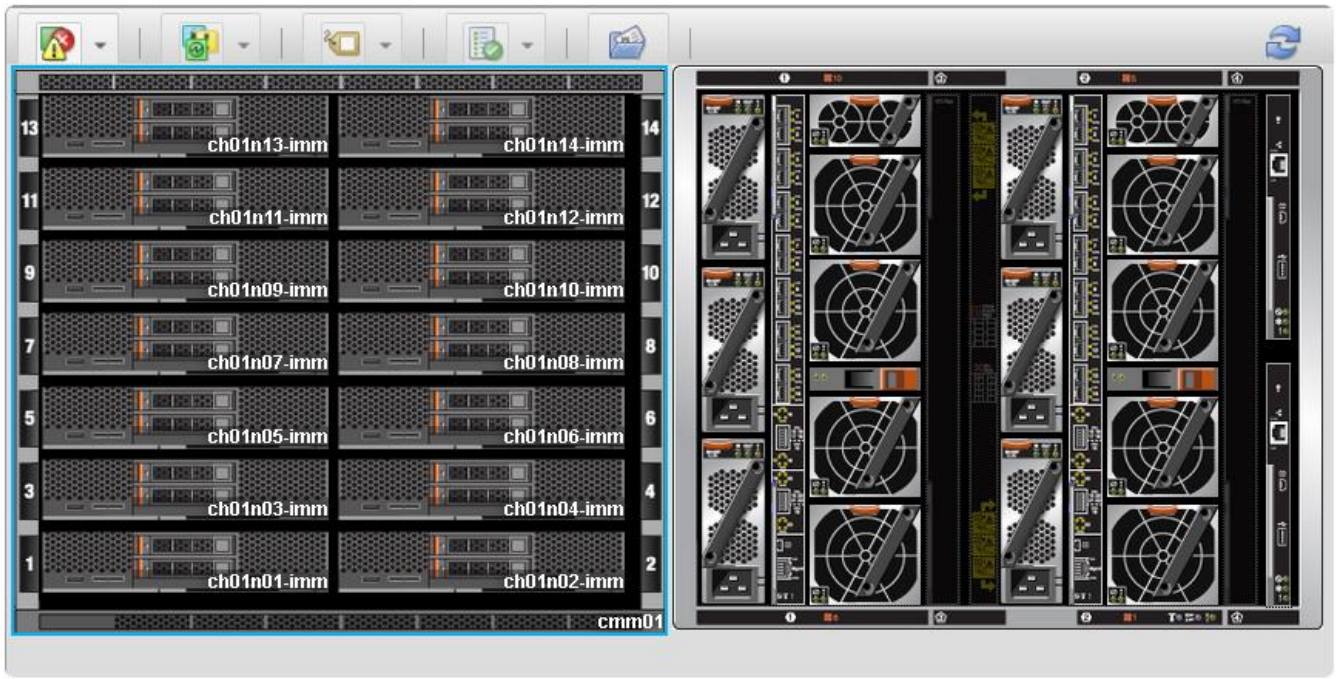

Lenovo XClarity Administrator
Dashboard
Hardware
Provisioning
Monitoring
Administration
Hardware Status
Servers
28
27
0
1
Storage
0
0
0
0
Switches
4
4
0
0
Chassis
2
1
0
1
Racks
1
0
0
1
Provisioning Status
Configuration Patterns
0 Servers with Profiles
28 Servers without Profiles
0 Server Pattern Deploys in Progress
Operating System Images
Operating System Images Not Ready
Firmware Updates
35 Devices Compliant
0 Devices Non-compliant
1 Devices without policy
0 Updates in Progress

Lenovo XClarity Administrator

Dashboard | Hardware | Provisioning | Monitoring | Administration

Chassis > cmm01

Graphic view | Table view



All Actions

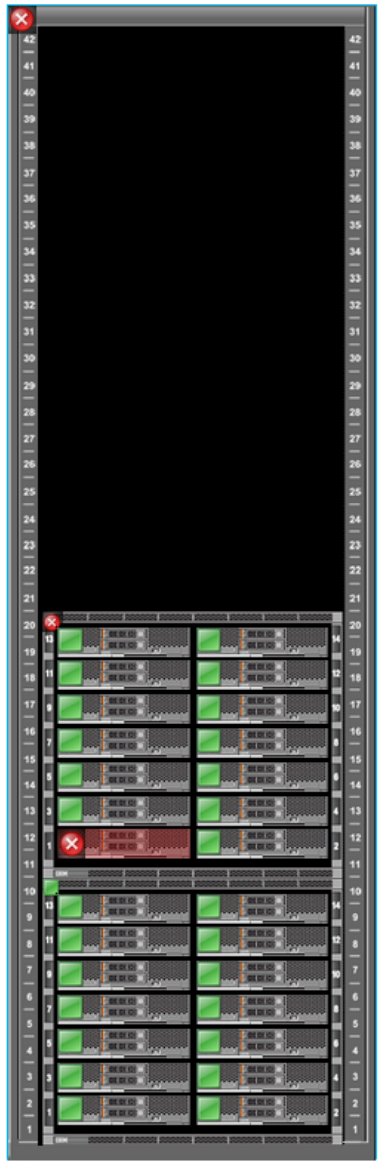
Summary **Details**

Name:	cmm01
Status:	Normal
Security Policy:	Secure
Host names:	MM00E0EC2C3D46
Serial number:	1006BEA
Type-Model:	8721-HC1

Lenovo XClarity Administrator

Dashboard | Hardware | Provisioning | Monitoring | Administration

All Racks > LCI-Rack-98fb472ec3

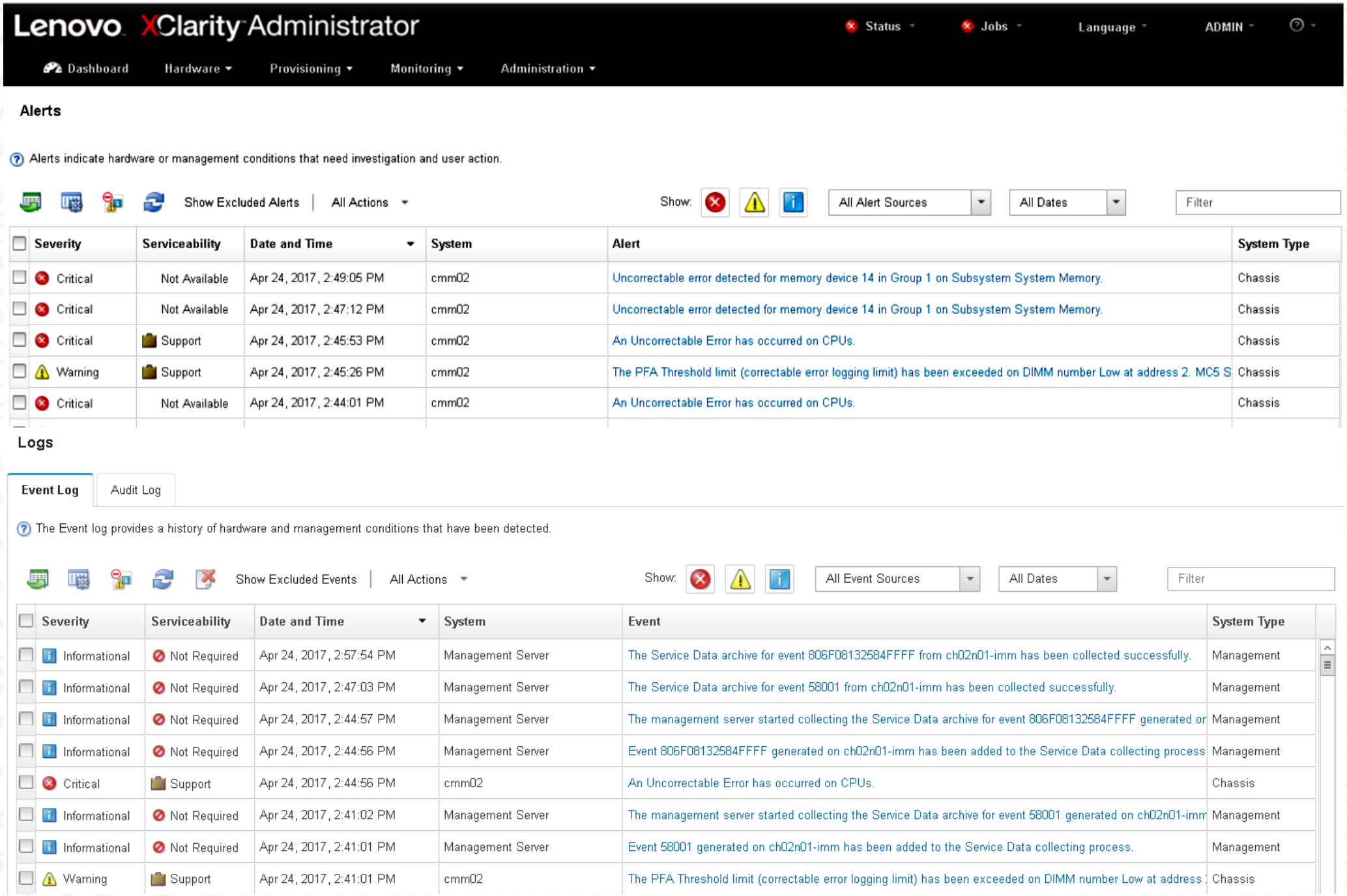


Edit Rack | All Actions

LCI-Rack-98fb472ec3 Edit Properties

Summary

Status:	Critical
Location:	
Room:	
Height:	42 units
Type:	Rack


Lenovo XClarity Administrator
Status
Jobs
Language
ADMIN
Dashboard
Hardware
Provisioning
Monitoring
Administration
Alerts
Alerts indicate hardware or management conditions that need investigation and user action.
Show Excluded Alerts
All Actions
Show:
All Alert Sources
All Dates
Filter
Severity
Serviceability
Date and Time
System
Alert
System Type
Critical
Not Available
Apr 24, 2017, 2:49:05 PM
cmm02
Uncorrectable error detected for memory device 14 in Group 1 on Subsystem System Memory.
Chassis
Critical
Not Available
Apr 24, 2017, 2:47:12 PM
cmm02
Uncorrectable error detected for memory device 14 in Group 1 on Subsystem System Memory.
Chassis
Critical
Support
Apr 24, 2017, 2:45:53 PM
cmm02
An Uncorrectable Error has occurred on CPUs.
Chassis
Warning
Support
Apr 24, 2017, 2:45:26 PM
cmm02
The PFA Threshold limit (correctable error logging limit) has been exceeded on DIMM number Low at address 2. MC5 S
Chassis
Critical
Not Available
Apr 24, 2017, 2:44:01 PM
cmm02
An Uncorrectable Error has occurred on CPUs.
Chassis
Logs
Event Log
Audit Log
The Event log provides a history of hardware and management conditions that have been detected.
Show Excluded Events
All Actions
Show:
All Event Sources
All Dates
Filter
Severity
Serviceability
Date and Time
System
Event
System Type
Informational
Not Required
Apr 24, 2017, 2:57:54 PM
Management Server
The Service Data archive for event 806F08132584FFFF from ch02n01-imm has been collected successfully.
Management
Informational
Not Required
Apr 24, 2017, 2:47:03 PM
Management Server
The Service Data archive for event 58001 from ch02n01-imm has been collected successfully.
Management
Informational
Not Required
Apr 24, 2017, 2:44:57 PM
Management Server
The management server started collecting the Service Data archive for event 806F08132584FFFF generated or
Management
Informational
Not Required
Apr 24, 2017, 2:44:56 PM
Management Server
Event 806F08132584FFFF generated on ch02n01-imm has been added to the Service Data collecting process
Management
Critical
Support
Apr 24, 2017, 2:44:56 PM
cmm02
An Uncorrectable Error has occurred on CPUs.
Chassis
Informational
Not Required
Apr 24, 2017, 2:41:02 PM
Management Server
The management server started collecting the Service Data archive for event 58001 generated on ch02n01-imm
Management
Informational
Not Required
Apr 24, 2017, 2:41:01 PM
Management Server
Event 58001 generated on ch02n01-imm has been added to the Service Data collecting process.
Management
Warning
Support
Apr 24, 2017, 2:41:01 PM
cmm02
The PFA Threshold limit (correctable error logging limit) has been exceeded on DIMM number Low at address
Chassis

More information

- Contact:
  - Srihari Angaluri sangaluri@lenovo.com
    Solution Architect
    Lenovo Data Center Group

- https://www.redhat.com/en/technologies/cloud-computing/openshift
- https://www.redhat.com/en/containers/what-is-kubernetes