Managing Kubernetes and OpenShift with ManageIQ

Alissa Bonas @ Container Con Seattle 2015

The stages of containers world

- Containerizing an app

The stages of containers world

- Run a container

The stages of containers world

- Run multiple containers

The stages of containers world

- Orchestrate containers
- Run many containers on multiple hosts
- Manage a containers environment

Today we’ll focus on

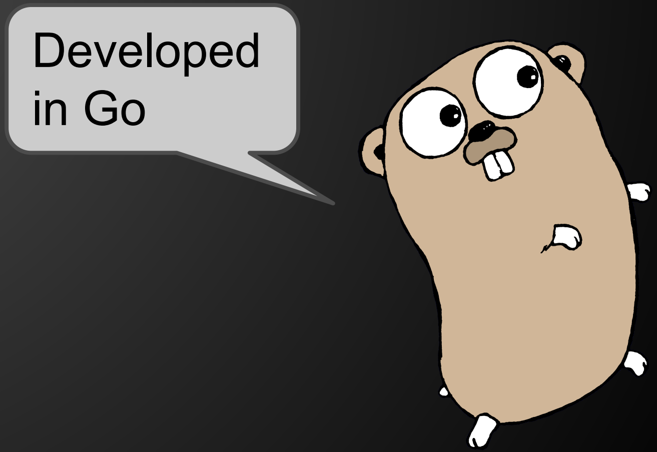
Kubernetes

OpenShift

ManageIQ

Kubernetes

- Deployment, scaling and orchestration of containers across clusters of hosts.
- http://kubernetes.io/
- 1.0 release - July 2015



Kubernetes main concepts

- Node - a machine that containers run on

- Namespace - partitioning resources created by users into logical groups

- Pod - a group of containers

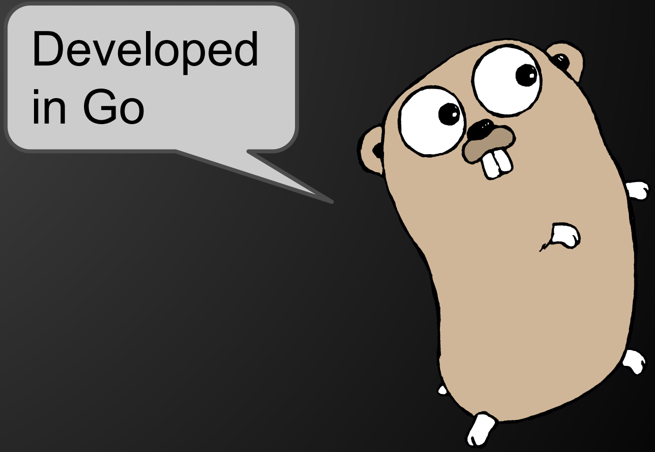
Kubernetes main concepts

- Replication controller - ensures there are always X replicas of pods

- Service - a base load balancer that provides traffic to pods

OpenShift

- Platform as a service for building and running applications - for developers
- http://www.openshift.org/
- 3.0 release - June 2015



OpenShift main concepts

- Built on top of Kubernetes

- Provides additional capabilities
  - application lifecycle
  - routing - extends service
  - projects - extension of namespaces

Build → Deploy → Run

Insight and control

- How many containers exist in my environment?
- Does a specific node have enough resources?
- How many distinct images are used?
- Which registries are used?

ManageIQ

- A cloud management platform
- http://manageiq.org/
- Botvinnik release - June 2015

a Ruby on Rails project

ManageIQ concepts

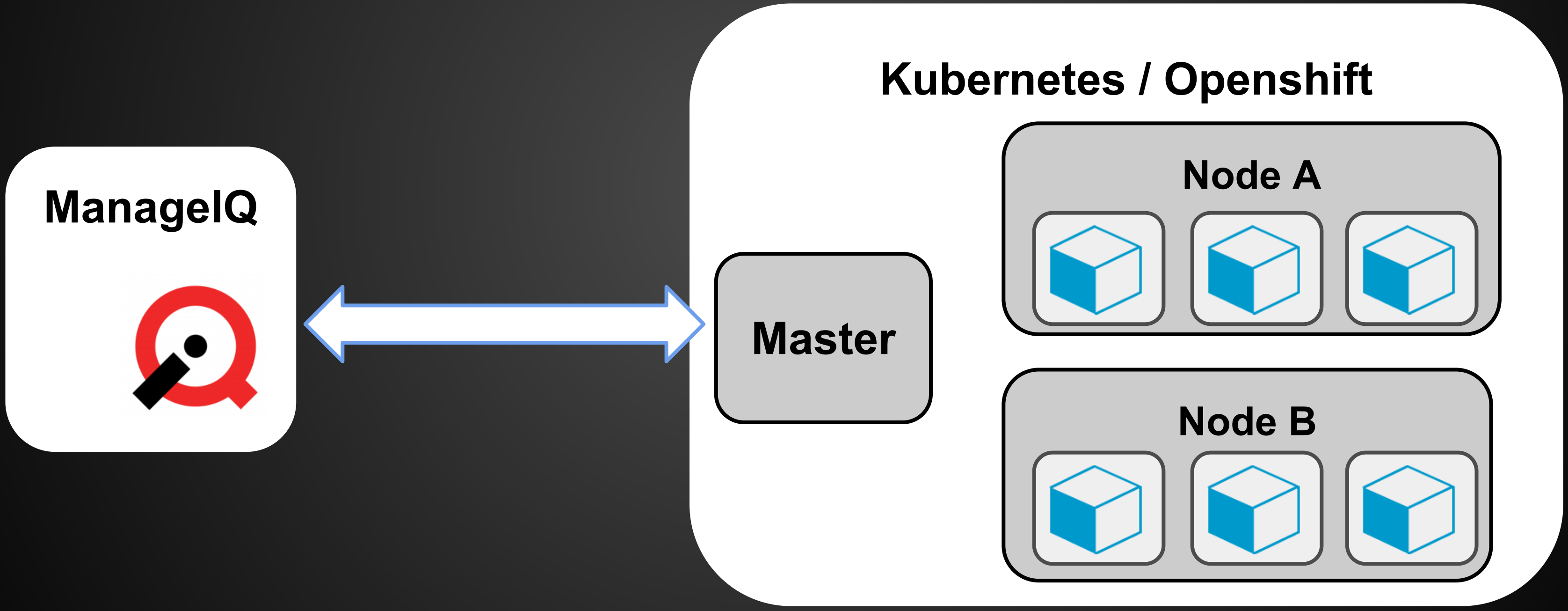
- “A manager of managers”
  - supports multiple virtualization providers
- Insight and control
  - inventory overview and events
  - smart state analysis
  - workflow/orchestration

New in upstream ManageIQ

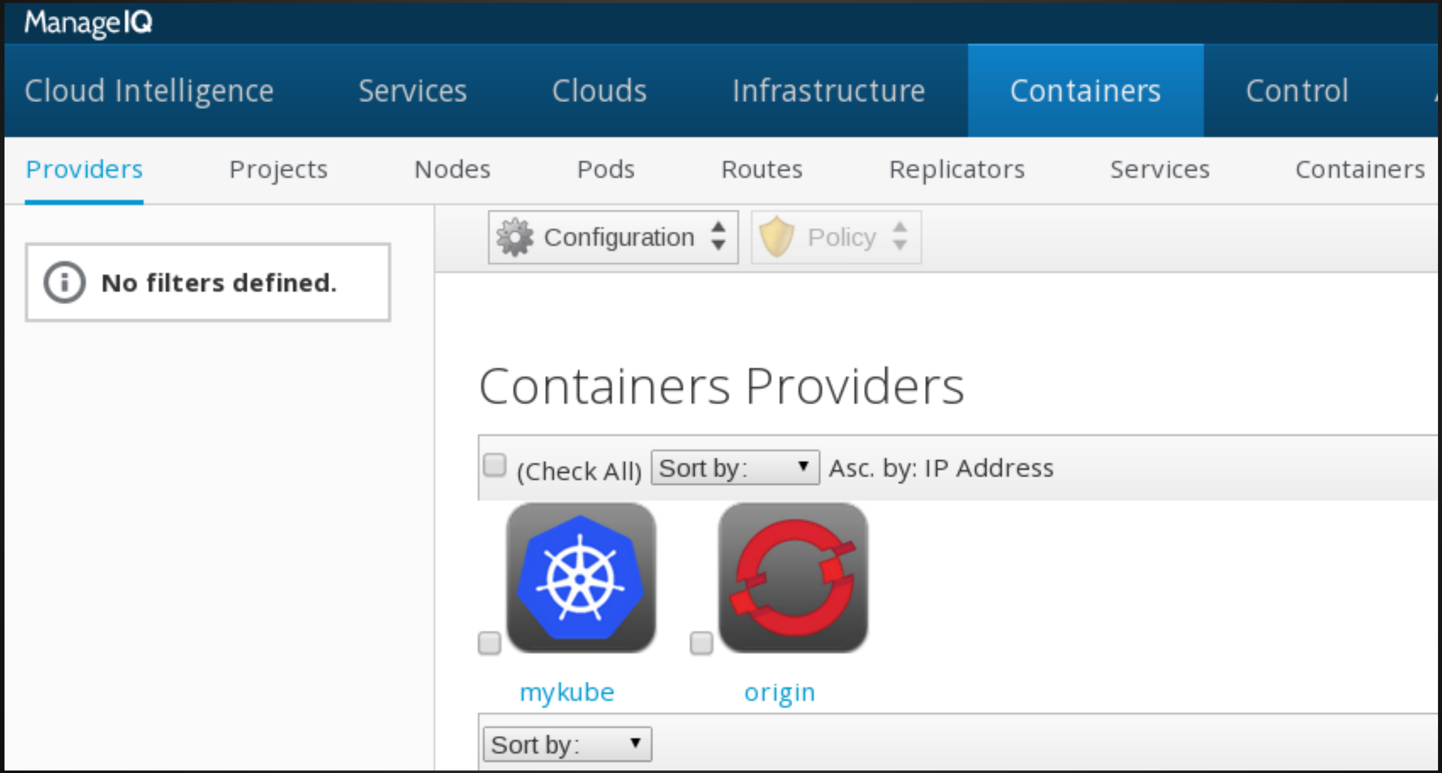
- Providers for container management

- Support added for
  - Kubernetes
  - OpenShift

Working together



Container management providers



No filters defined.

Containers Providers

□ (Check All) Sort by: ▾ Asc. by: IP Address

□ mykube

□ origin

Sort by: ▾

Creating a provider

- Kubernetes/OpenShift master host/port
- SSL
- Token based

Inventory

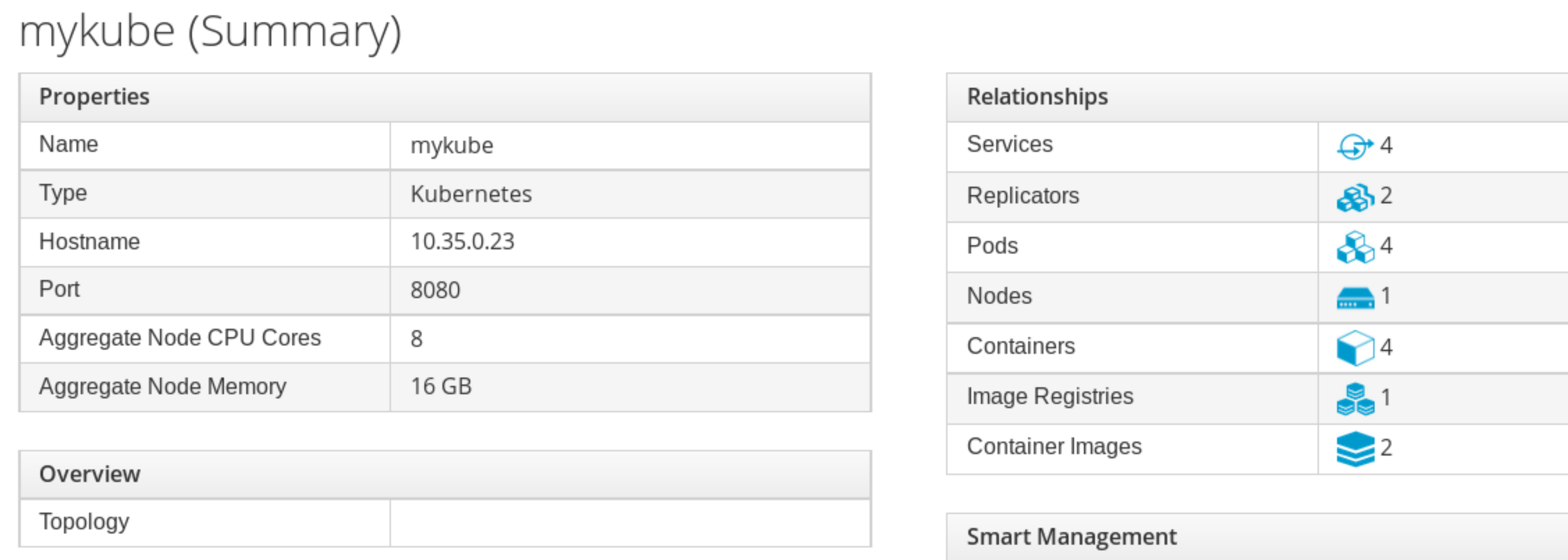
- Entities
- Relationships
- Additional information

Creating more insights

- Modelling additional entities as first class citizens

- Deducing relationships

Kubernetes provider summary page



mykube (Summary)

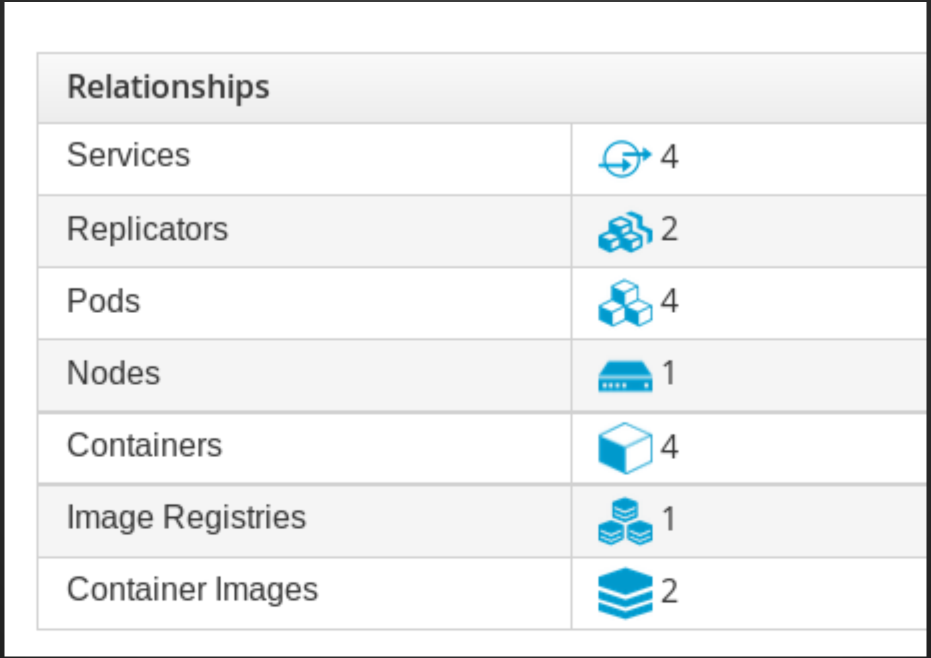
Properties	
Name	mykube
Type	Kubernetes
Hostname	10.35.0.23
Port	8080
Aggregate Node CPU Cores	8
Aggregate Node Memory	16 GB

Overview	
Topology	

Relationships	
Services	4
Replicators	2
Pods	4
Nodes	1
Containers	4
Image Registries	1
Container Images	2

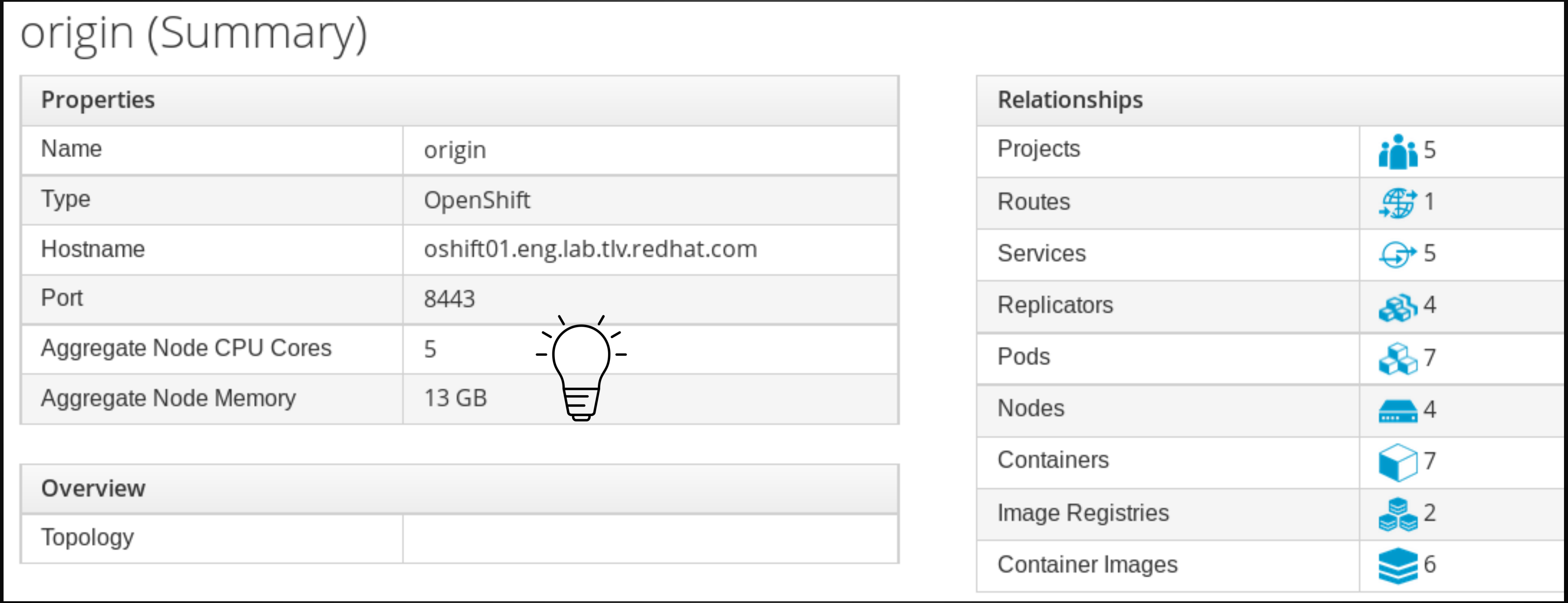
Smart Management

Kubernetes provider relationships



Relationships	
Services	4
Replicators	2
Pods	4
Nodes	1
Containers	4
Image Registries	1
Container Images	2

OpenShift provider summary page



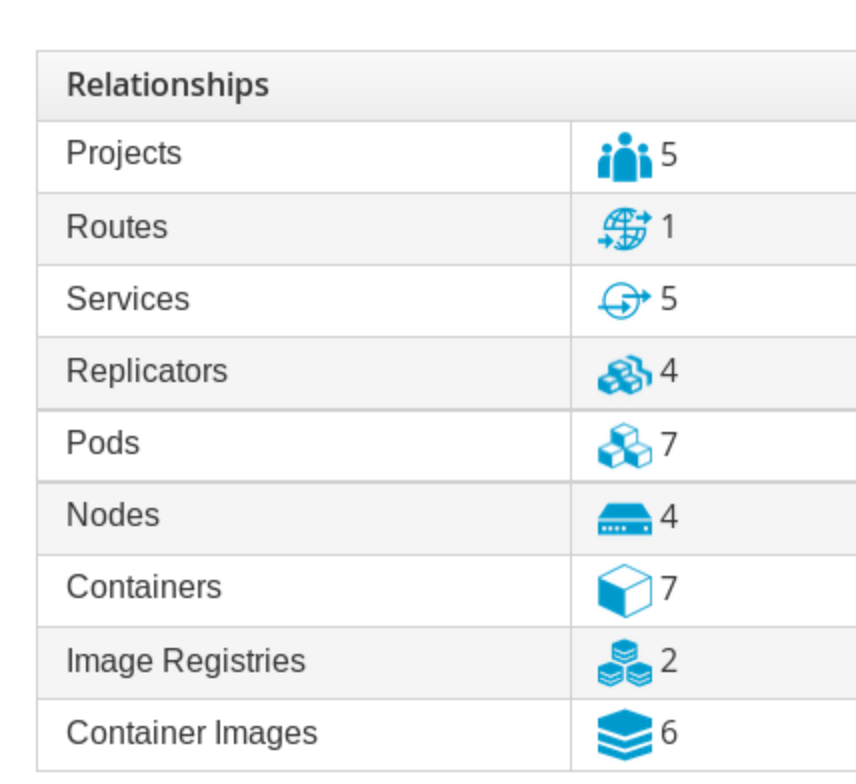
origin (Summary)

Properties	
Name	origin
Type	OpenShift
Hostname	oshift01.eng.lab.tlv.redhat.com
Port	8443
Aggregate Node CPU Cores	5
Aggregate Node Memory	13 GB

Overview	
Topology	

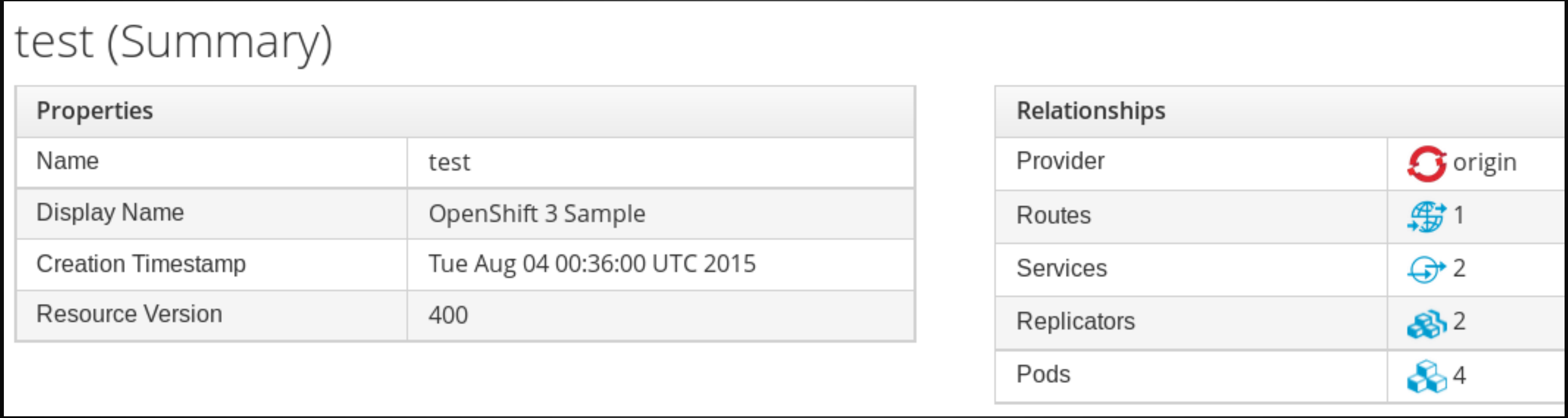
Relationships	
Projects	5
Routes	1
Services	5
Replicators	4
Pods	7
Nodes	4
Containers	7
Image Registries	2
Container Images	6

OpenShift provider relationships



Relationships	
Projects	5
Routes	1
Services	5
Replicators	4
Pods	7
Nodes	4
Containers	7
Image Registries	2
Container Images	6

Projects



test (Summary)

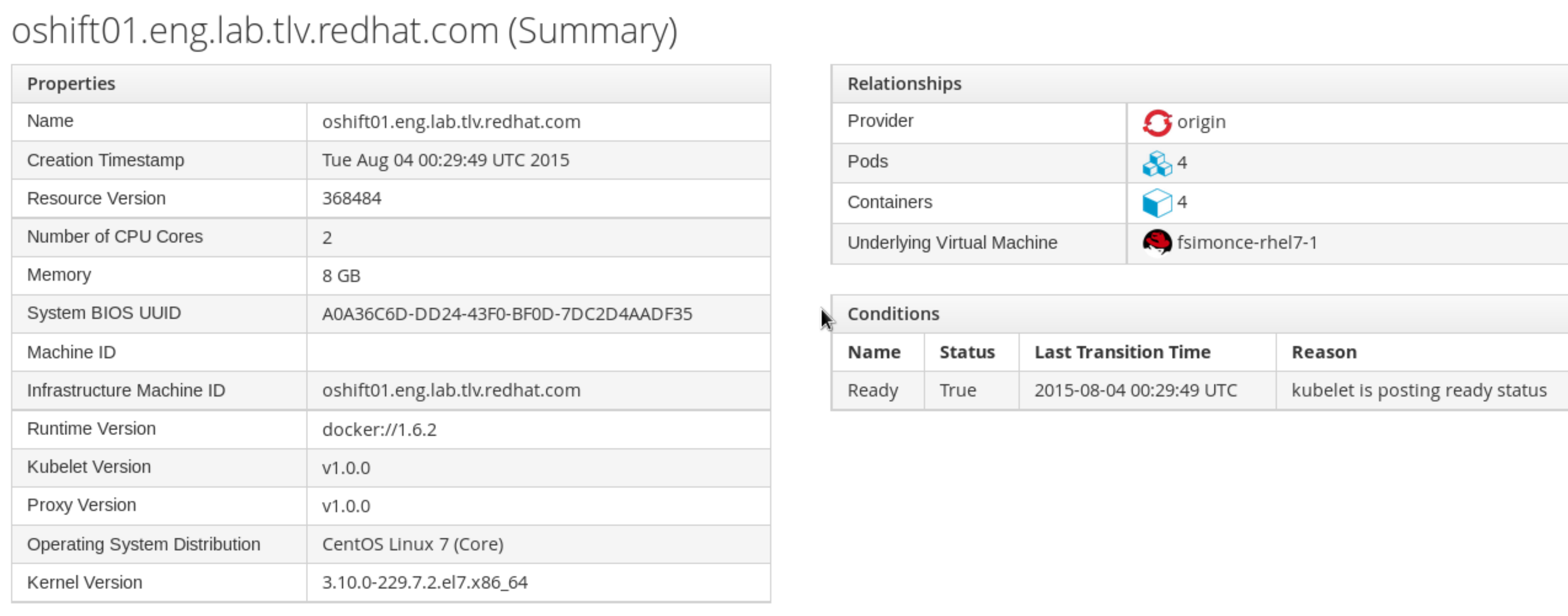
Properties	
Name	test
Display Name	OpenShift 3 Sample
Creation Timestamp	Tue Aug 04 00:36:00 UTC 2015
Resource Version	400

Relationships	
Provider	origin
Routes	1
Services	2
Replicators	2
Pods	4

Nodes

- OS and Software versions
- How many entities are on a node
- Capacity and utilization
- Which infrastructure is it running on

Node Summary page



oshift01.eng.lab.tlv.redhat.com (Summary)

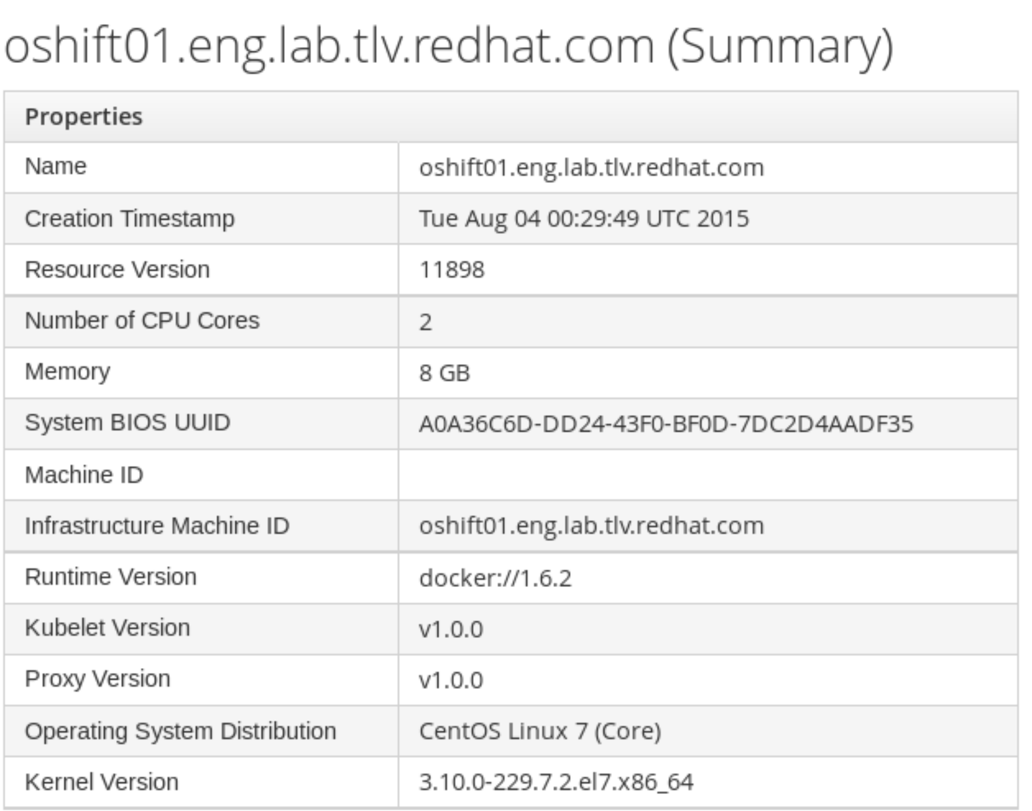
Properties	
Name	oshift01.eng.lab.tlv.redhat.com
Creation Timestamp	Tue Aug 04 00:29:49 UTC 2015
Resource Version	368484
Number of CPU Cores	2
Memory	8 GB
System BIOS UUID	A0A36C6D-DD24-43F0-BF0D-7DC2D4AADF35
Machine ID	
Infrastructure Machine ID	oshift01.eng.lab.tlv.redhat.com
Runtime Version	docker://1.6.2
Kubelet Version	v1.0.0
Proxy Version	v1.0.0
Operating System Distribution	CentOS Linux 7 (Core)
Kernel Version	3.10.0-229.7.2.el7.x86_64

Relationships	
Provider	origin
Pods	4
Containers	4
Underlying Virtual Machine	fsimonce-rhel7-1

Conditions

Name	Status	Last Transition Time	Reason
Ready	True	2015-08-04 00:29:49 UTC	kubelet is posting ready status

What do we know about nodes?



oshift01.eng.lab.tlv.redhat.com (Summary)

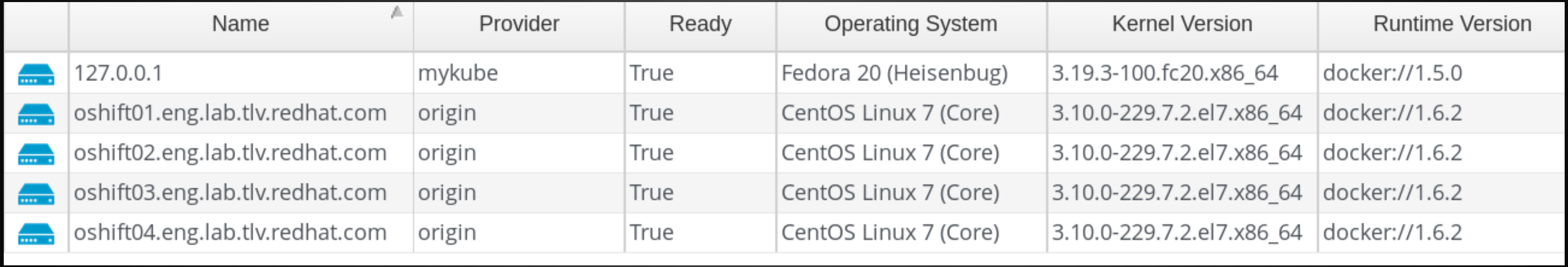
Properties	
Name	oshift01.eng.lab.tlv.redhat.com
Creation Timestamp	Tue Aug 04 00:29:49 UTC 2015
Resource Version	11898
Number of CPU Cores	2
Memory	8 GB
System BIOS UUID	A0A36C6D-DD24-43F0-BF0D-7DC2D4AADF35
Machine ID	
Infrastructure Machine ID	oshift01.eng.lab.tlv.redhat.com
Runtime Version	docker://1.6.2
Kubelet Version	v1.0.0
Proxy Version	v1.0.0
Operating System Distribution	CentOS Linux 7 (Core)
Kernel Version	3.10.0-229.7.2.el7.x86_64

Capacity

Docker and Kubernetes info

Nodes

- Quick overview on main nodes properties



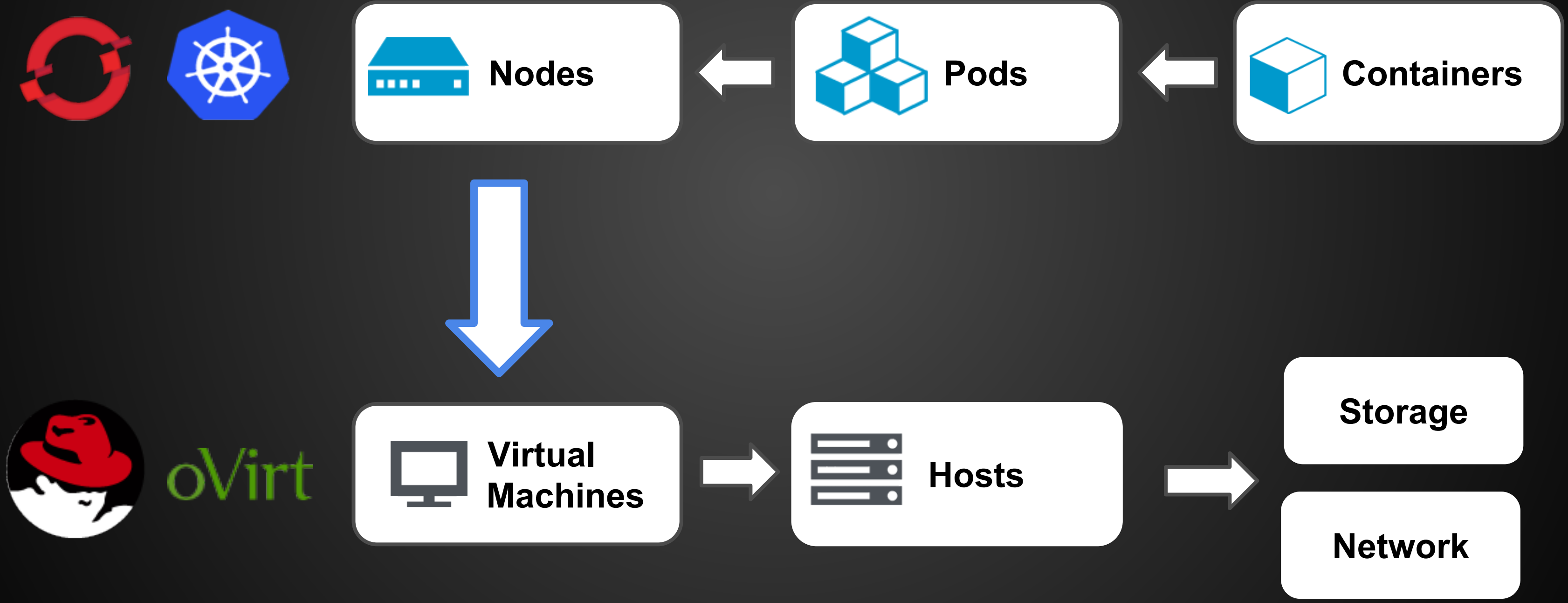
	Name	Provider	Ready	Operating System	Kernel Version	Runtime Version
	127.0.0.1	mykube	True	Fedora 20 (Heisenbug)	3.19.3-100.fc20.x86_64	docker://1.5.0
	oshift01.eng.lab.tlv.redhat.com	origin	True	CentOS Linux 7 (Core)	3.10.0-229.7.2.el7.x86_64	docker://1.6.2
	oshift02.eng.lab.tlv.redhat.com	origin	True	CentOS Linux 7 (Core)	3.10.0-229.7.2.el7.x86_64	docker://1.6.2
	oshift03.eng.lab.tlv.redhat.com	origin	True	CentOS Linux 7 (Core)	3.10.0-229.7.2.el7.x86_64	docker://1.6.2
	oshift04.eng.lab.tlv.redhat.com	origin	True	CentOS Linux 7 (Core)	3.10.0-229.7.2.el7.x86_64	docker://1.6.2

Cross providers insight

- Connect all layers of infrastructure, cloud and containers

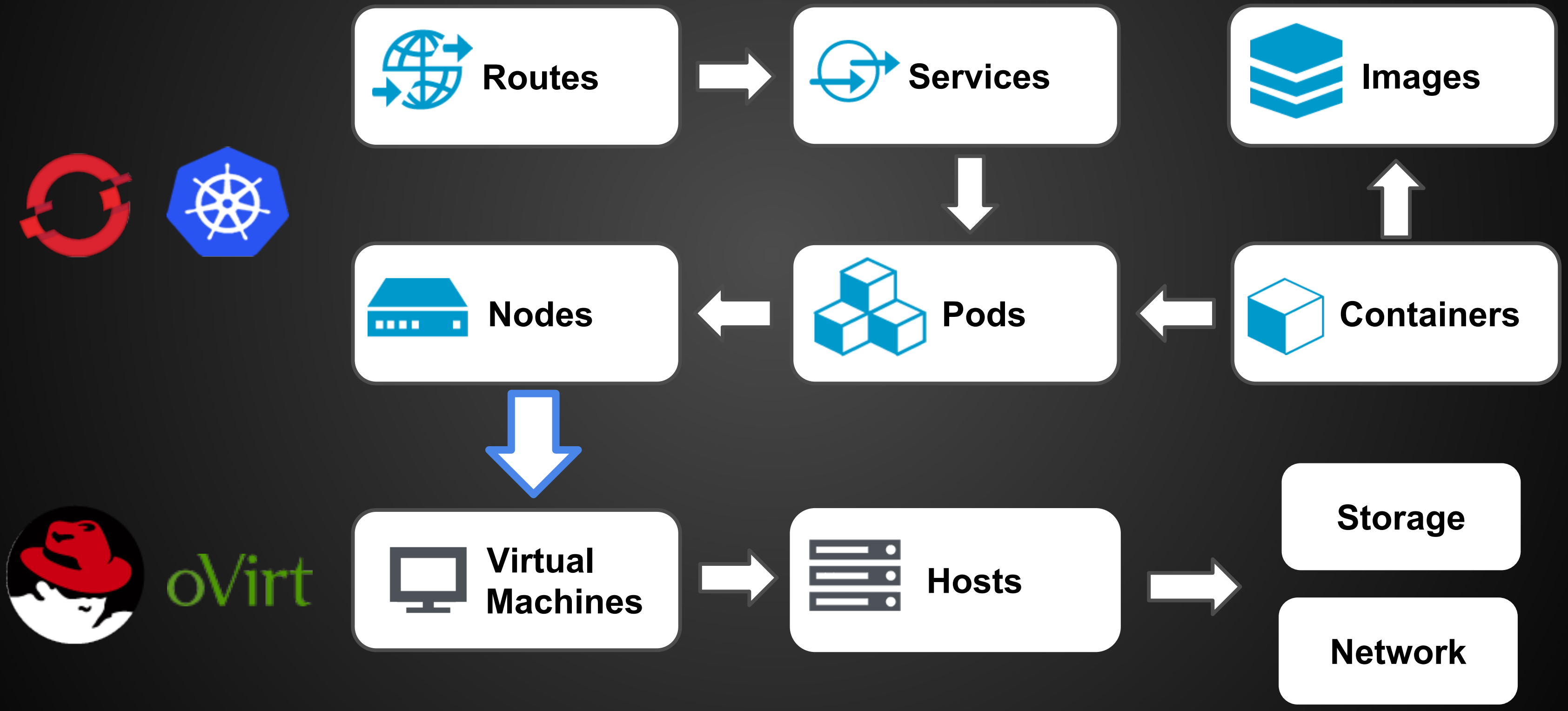
- Currently support cross linking with
  - OpenStack
  - oVirt / RHEV

Cross linking demystified



Alissa Bonas @ Container Con Seattle 2015

Cross linking demystified



Routes

Services

Images

Nodes

Pods

Containers

Virtual Machines

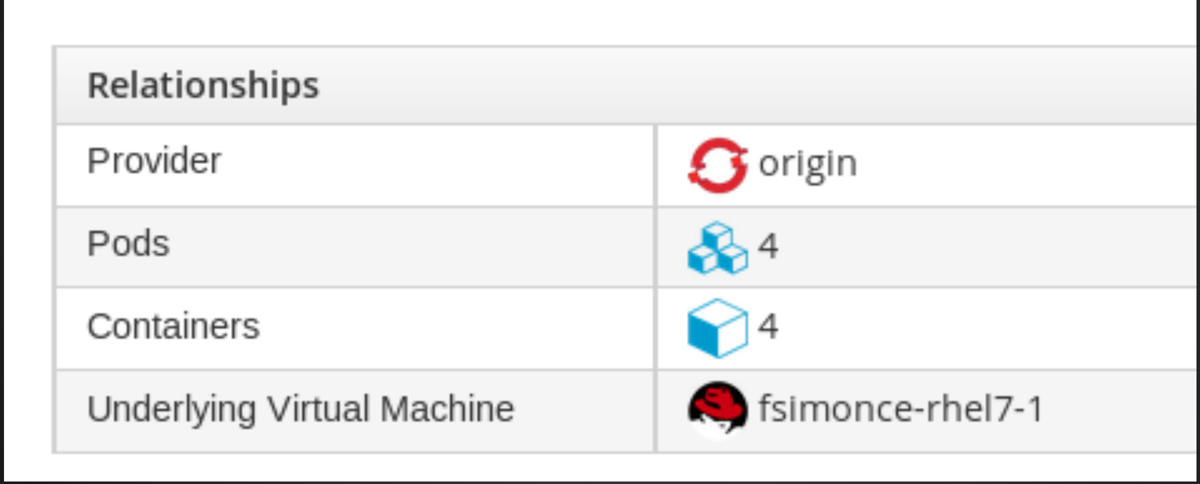
Hosts

Storage

Network

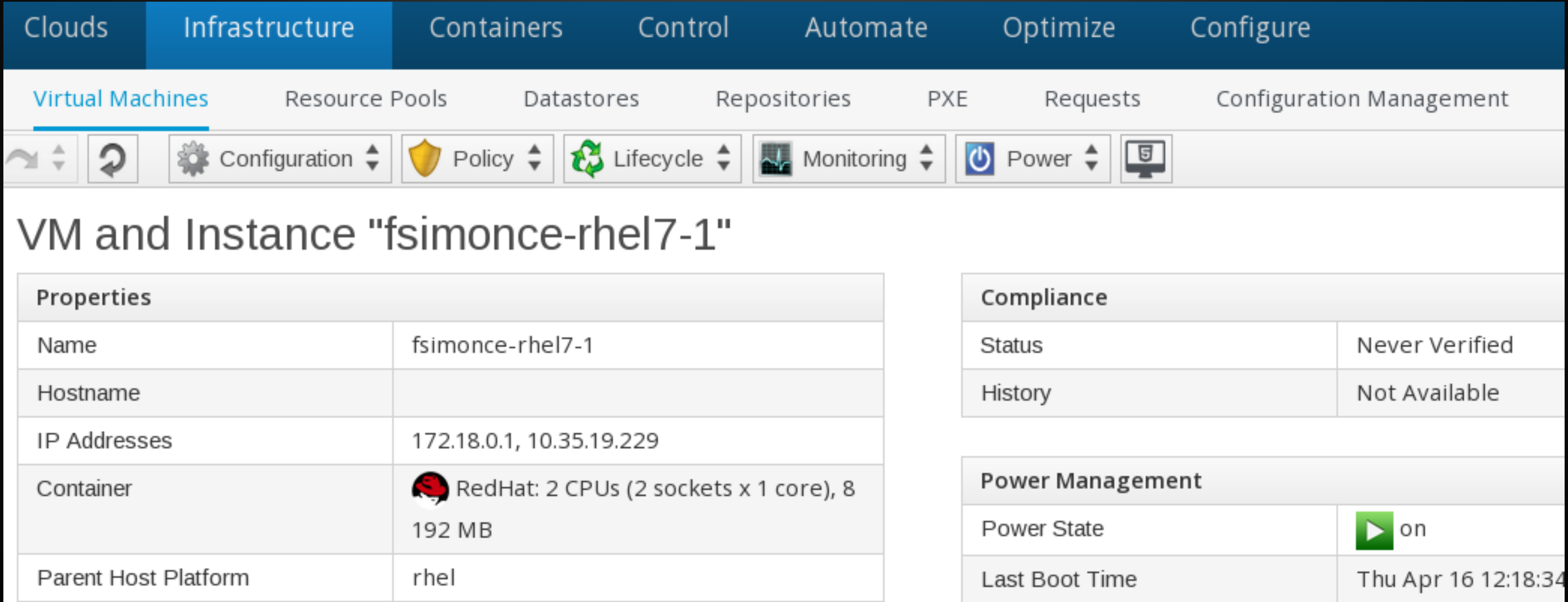
Alissa Bonas @ Container Con Seattle 2015

Cross provider example - Node



Relationships	
Provider	origin
Pods	4
Containers	4
Underlying Virtual Machine	fsimonce-rhel7-1

Cross provider VM example



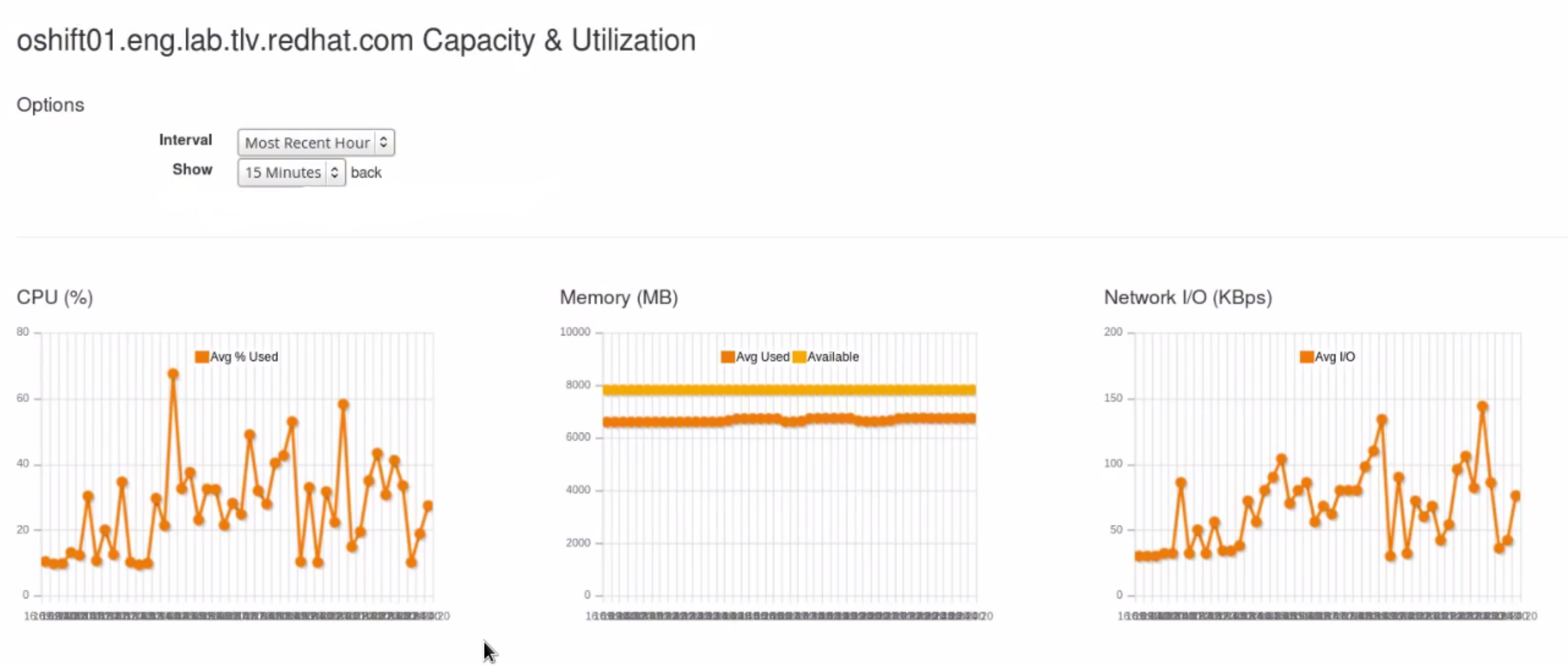
VM and Instance "fsimonce-rhel7-1"

Properties	
Name	fsimonce-rhel7-1
Hostname	
IP Addresses	172.18.0.1, 10.35.19.229
Container	RedHat: 2 CPUs (2 sockets x 1 core), 8 192 MB
Parent Host Platform	rhel

Compliance	
Status	Never Verified
History	Not Available

Power Management	
Power State	▶ on
Last Boot Time	Thu Apr 16 12:18:34

Nodes capacity and utilization



oshift01.eng.lab.tlv.redhat.com Capacity & Utilization

Options

Interval: Most Recent Hour

Show: 15 Minutes back

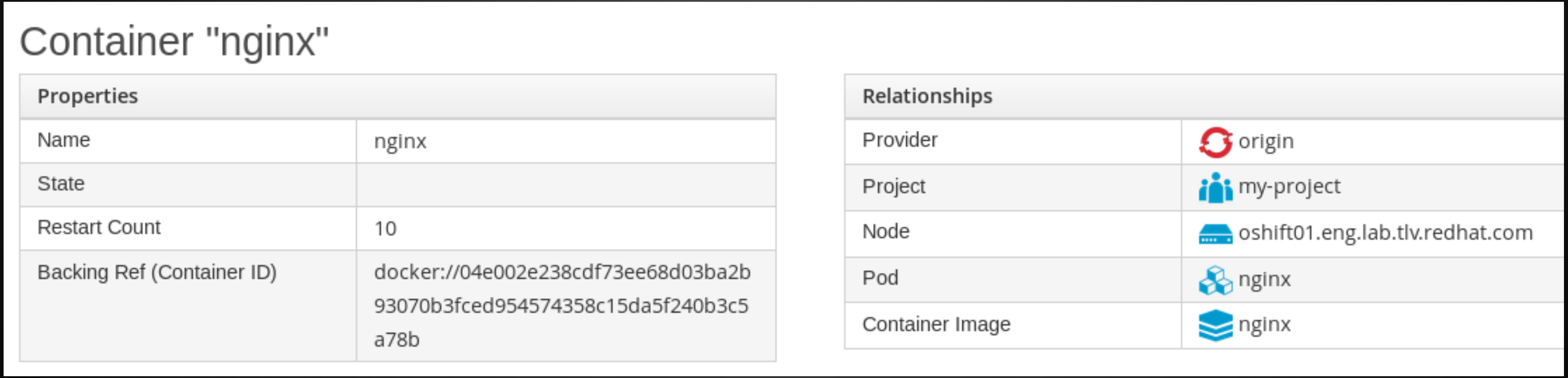
Resource quotas and limit ranges

- Limit the number of pods, containers, etc.
    - tracked per a project/namespace scope

- Limit CPU and memory
    - tracked per pod, container

Container

- Traceability - container id, image
- Restart count - potential issues?

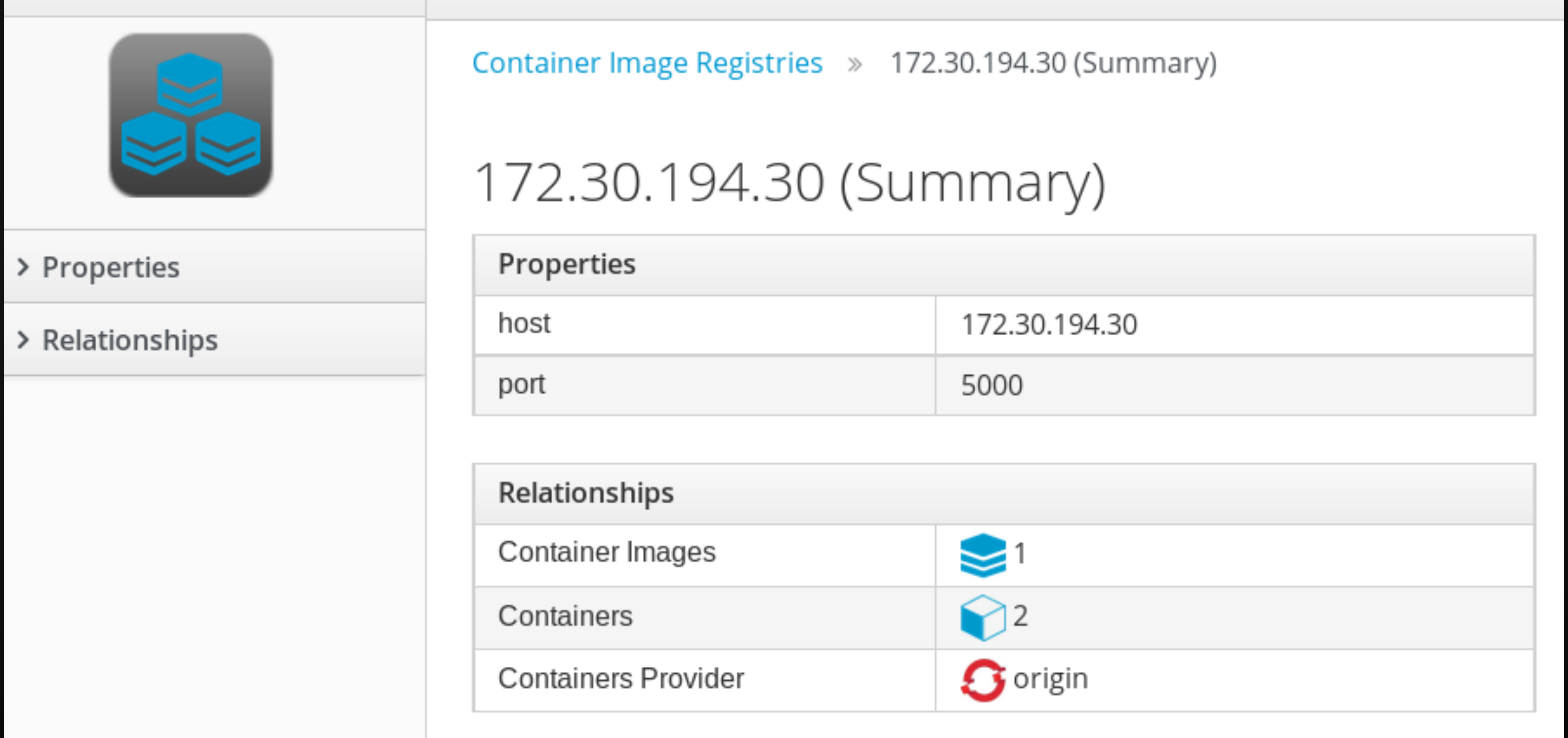


Container "nginx"

Properties	
Name	nginx
State	
Restart Count	10
Backing Ref (Container ID)	docker://04e002e238cdf73ee68d03ba2b93070b3fced954574358c15da5f240b3c5a78b

Relationships	
Provider	origin
Project	my-project
Node	oshift01.eng.lab.tlv.redhat.com
Pod	nginx
Container Image	nginx

Registries



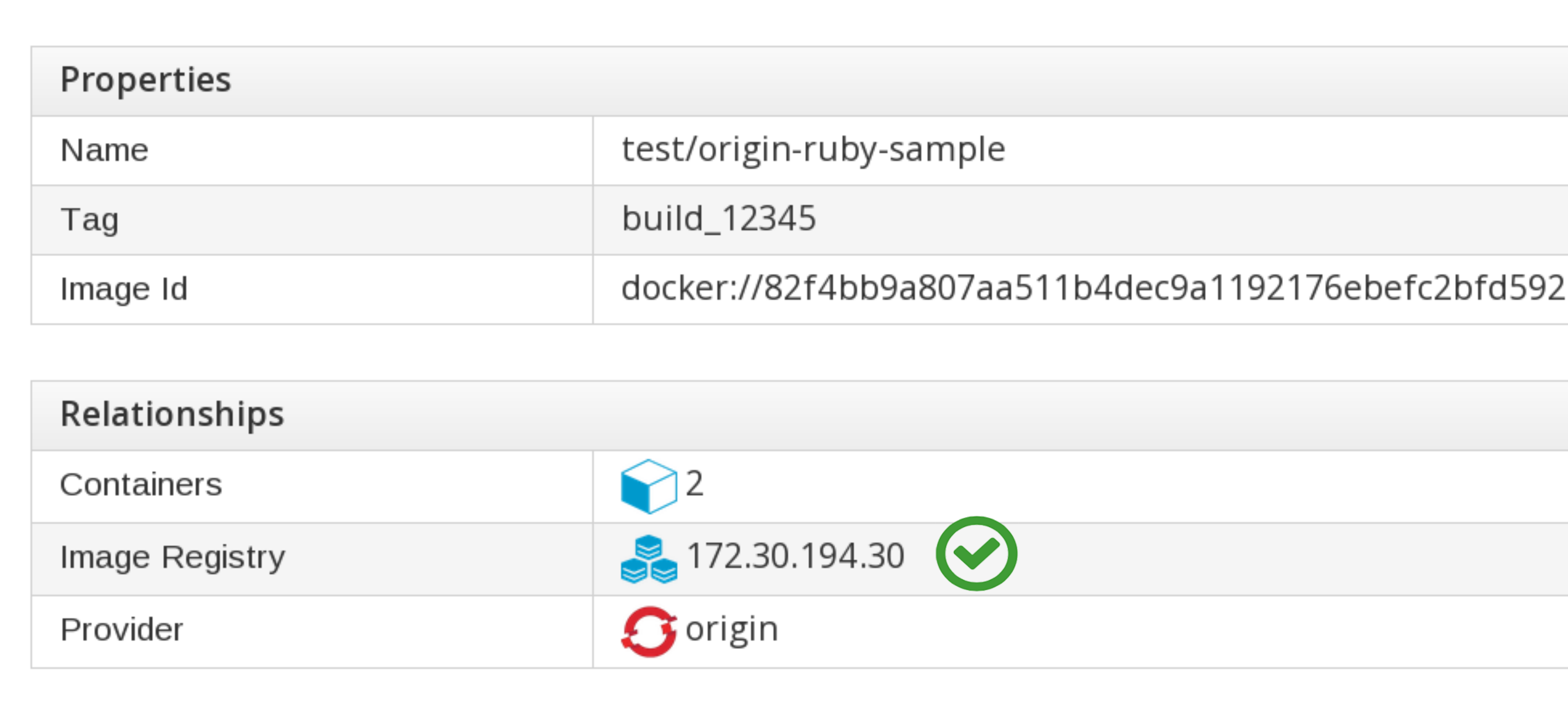
Container Image Registries » 172.30.194.30 (Summary)

172.30.194.30 (Summary)

Properties	
host	172.30.194.30
port	5000

Relationships	
Container Images	1
Containers	2
Containers Provider	origin

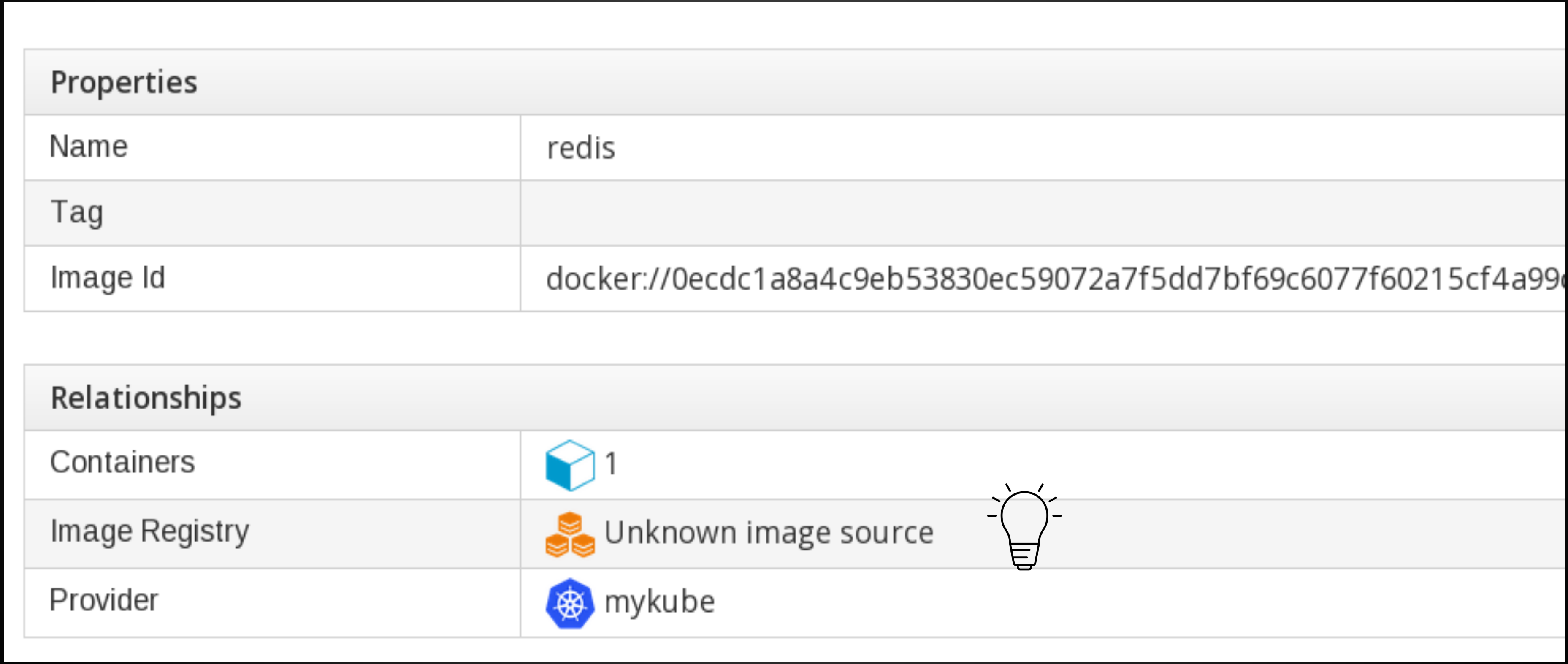
Image from a known registry



Properties	
Name	test/origin-ruby-sample
Tag	build_12345
Image Id	docker://82f4bb9a807aa511b4dec9a1192176ebefc2bfd592

Relationships	
Containers	2
Image Registry	172.30.194.30 ✓
Provider	origin

Image from an unknown source



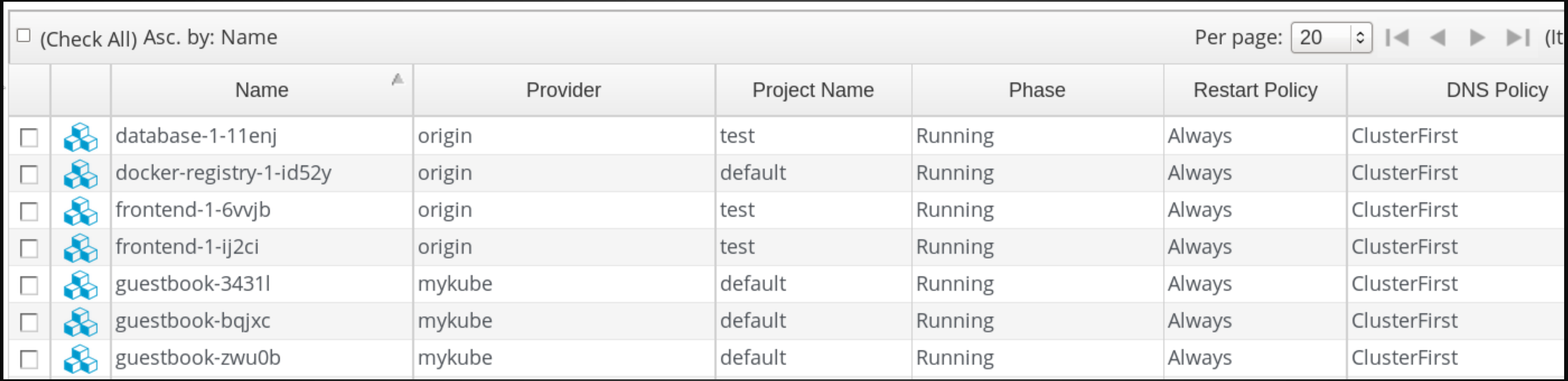
Properties	
Name	redis
Tag	
Image Id	docker://0ecdc1a8a4c9eb53830ec59072a7f5dd7bf69c6077f60215cf4a99

Relationships	
Containers	1
Image Registry	Unknown image source
Provider	mykube

Pods

- Which containers are part of it
- Which services work with it
- Which node does it run on
- Is it controlled by a replicator?

Pods

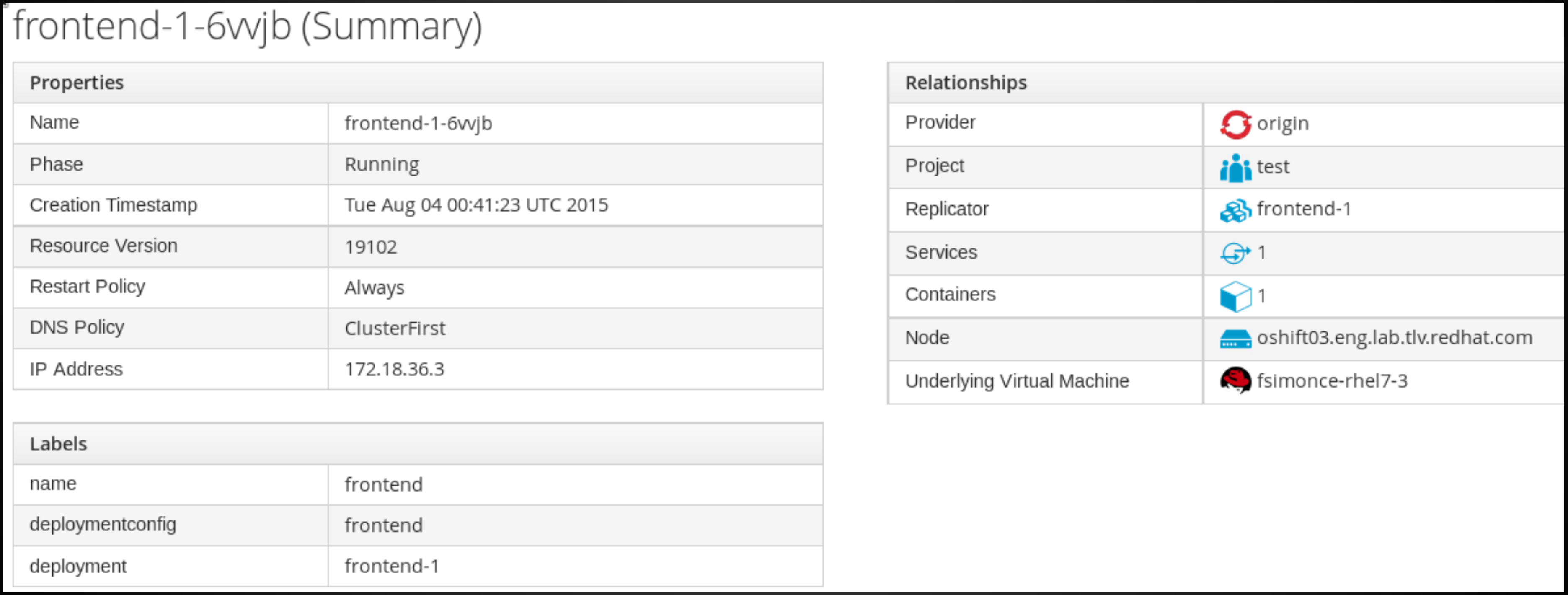


□ (Check All) Asc. by: Name

Per page: 20 (It

		Name	Provider	Project Name	Phase	Restart Policy	DNS Policy
□		database-1-11enj	origin	test	Running	Always	ClusterFirst
□		docker-registry-1-id52y	origin	default	Running	Always	ClusterFirst
□		frontend-1-6vvjb	origin	test	Running	Always	ClusterFirst
□		frontend-1-ij2ci	origin	test	Running	Always	ClusterFirst
□		guestbook-3431l	mykube	default	Running	Always	ClusterFirst
□		guestbook-bqjxc	mykube	default	Running	Always	ClusterFirst
□		guestbook-zwu0b	mykube	default	Running	Always	ClusterFirst

Pod



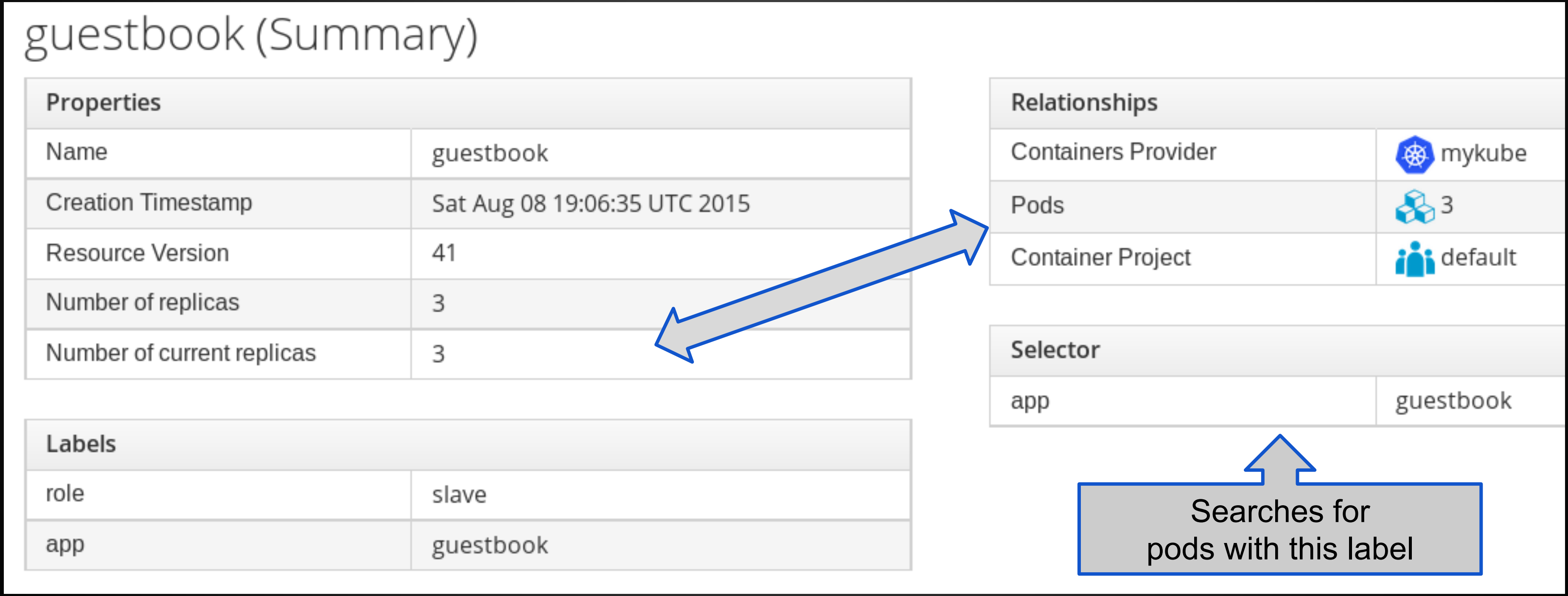
frontend-1-6vvjb (Summary)

Properties	
Name	frontend-1-6vvjb
Phase	Running
Creation Timestamp	Tue Aug 04 00:41:23 UTC 2015
Resource Version	19102
Restart Policy	Always
DNS Policy	ClusterFirst
IP Address	172.18.36.3

Labels	
name	frontend
deploymentconfig	frontend
deployment	frontend-1

Relationships	
Provider	origin
Project	test
Replicator	frontend-1
Services	1
Containers	1
Node	oshift03.eng.lab.tlv.redhat.com
Underlying Virtual Machine	fsimonce-rhel7-3

Replicators



guestbook (Summary)

Properties	
Name	guestbook
Creation Timestamp	Sat Aug 08 19:06:35 UTC 2015
Resource Version	41
Number of replicas	3
Number of current replicas	3

Labels	
role	slave
app	guestbook

Relationships	
Containers Provider	mykube
Pods	3
Container Project	default

Selector	
app	guestbook

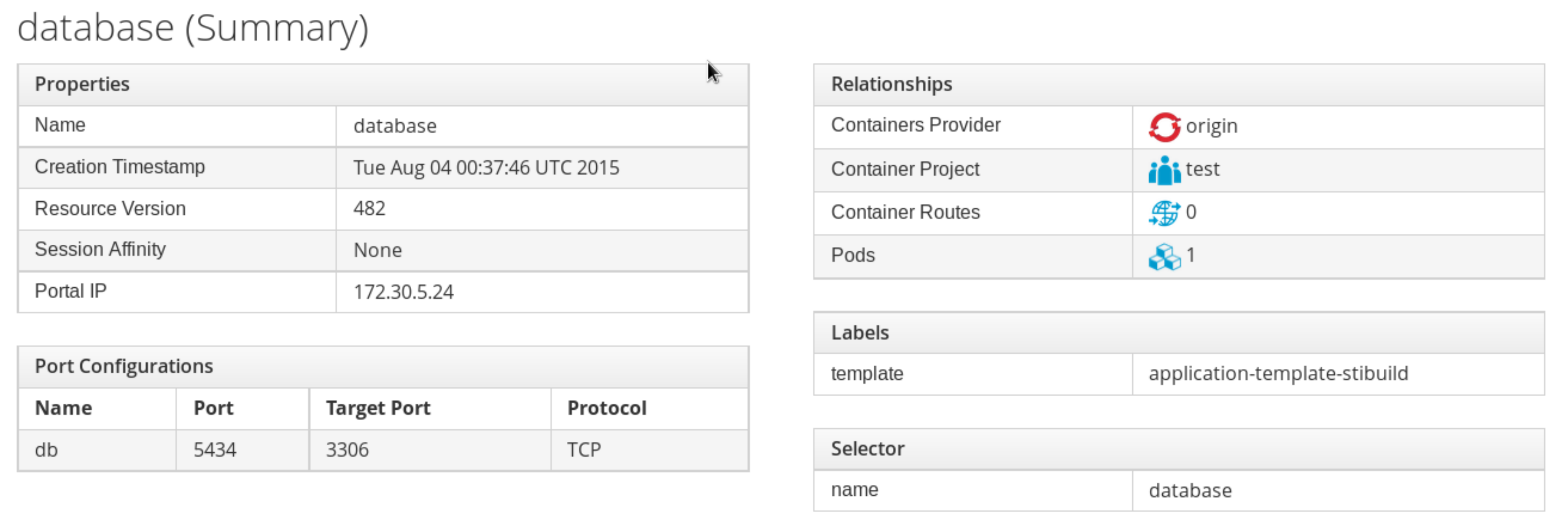
Searches for
pods with this label

Services

- A portal IP and source/target port pairs

- Redirects traffic to relevant pods based on a labels selector

Service example



database (Summary)

Properties	
Name	database
Creation Timestamp	Tue Aug 04 00:37:46 UTC 2015
Resource Version	482
Session Affinity	None
Portal IP	172.30.5.24

Port Configurations

Name	Port	Target Port	Protocol
db	5434	3306	TCP

Relationships	
Containers Provider	origin
Container Project	test
Container Routes	0
Pods	1

Labels	
template	application-template-stibuild

Selector	
name	database

OpenShift Routes

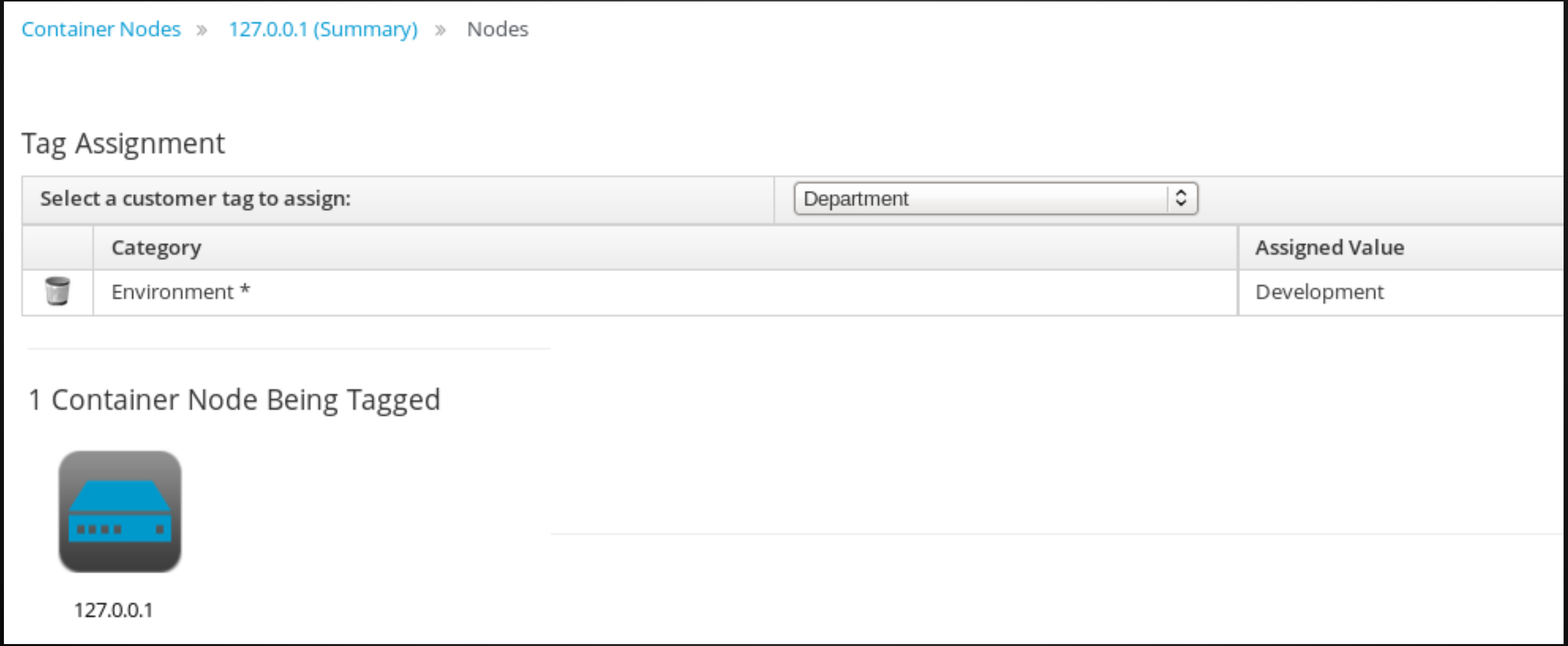
- Exposes a service by giving it an externally reachable hostname
- Can be fine tuned by /path
- Can be also secured

Events / Timeline

- Node
    - ready / not ready / rebooted
- Pod
    - scheduled
- More to come...

Tagging

- Leveraging ManageIQ tags



Tag Assignment

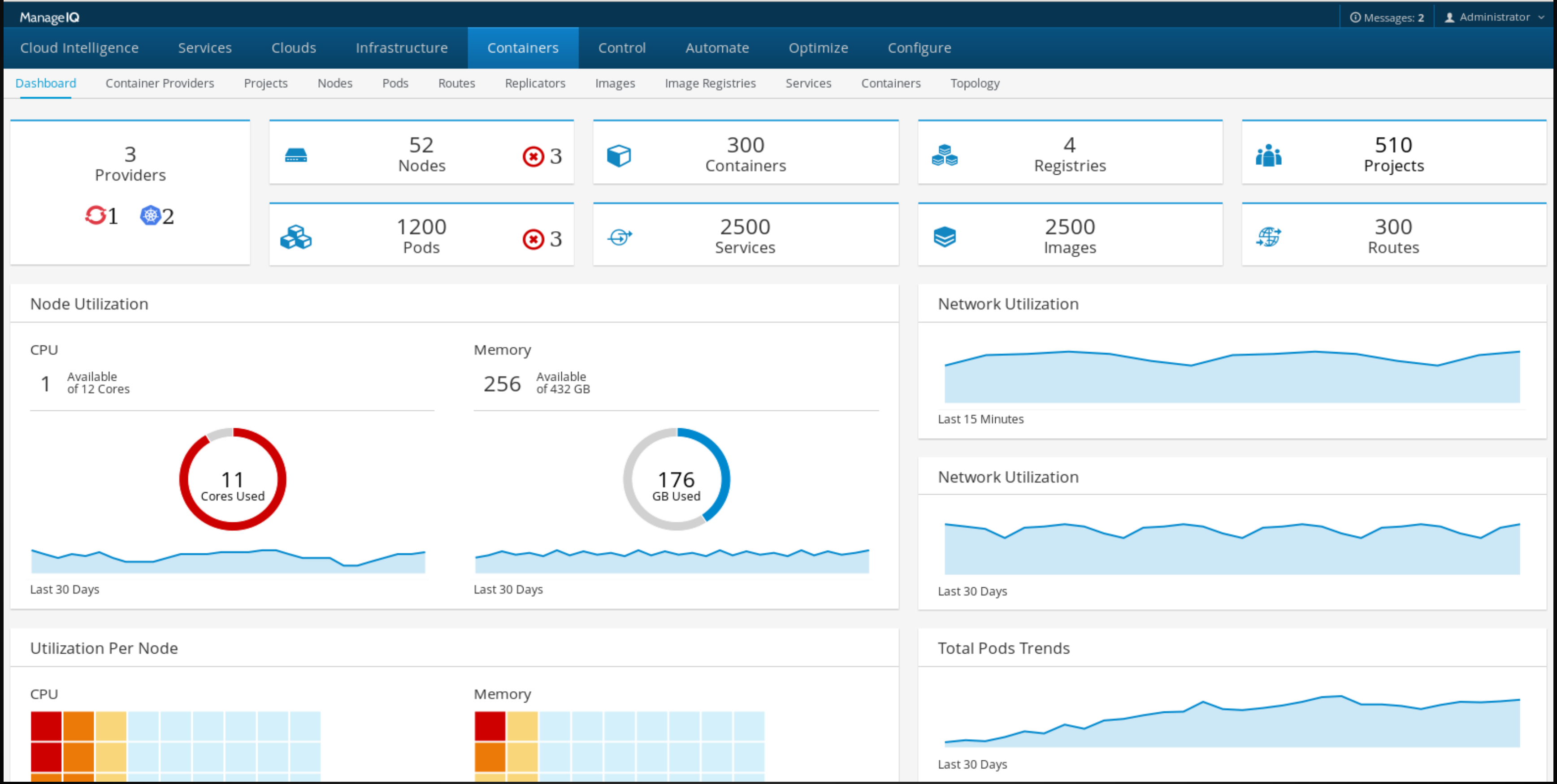
Select a customer tag to assign: Department

	Category	Assigned Value
🗑	Environment *	Development

1 Container Node Being Tagged

A glimpse into the future

Dashboard - providers overview



3 Providers
1 2

52 Nodes 3

300 Containers

4 Registries

510 Projects

1200 Pods 3

2500 Services

2500 Images

300 Routes

Node Utilization

CPU
1 Available of 12 Cores
11 Cores Used
Last 30 Days

Memory
256 Available of 432 GB
176 GB Used
Last 30 Days

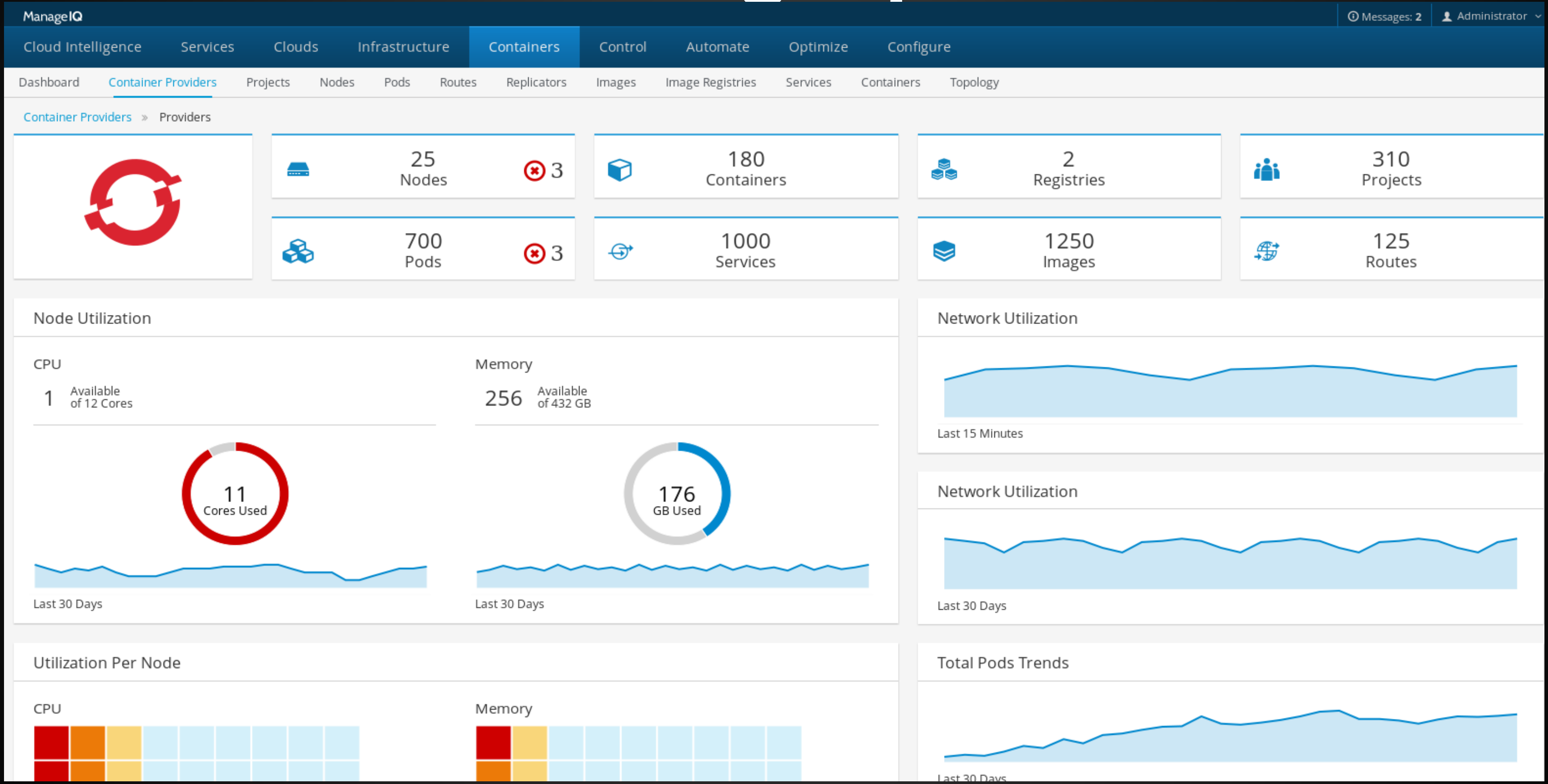
Network Utilization
Last 15 Minutes

Network Utilization
Last 30 Days

Utilization Per Node
CPU
Memory

Total Pods Trends
Last 30 Days

Dashboard - a single provider view



25 Nodes ⊗ 3

180 Containers

2 Registries

310 Projects

700 Pods ⊗ 3

1000 Services

1250 Images

125 Routes

Node Utilization

CPU

1 Available of 12 Cores

11 Cores Used

Last 30 Days

Memory

256 Available of 432 GB

176 GB Used

Last 30 Days

Network Utilization

Last 15 Minutes

Network Utilization

Last 30 Days

Utilization Per Node

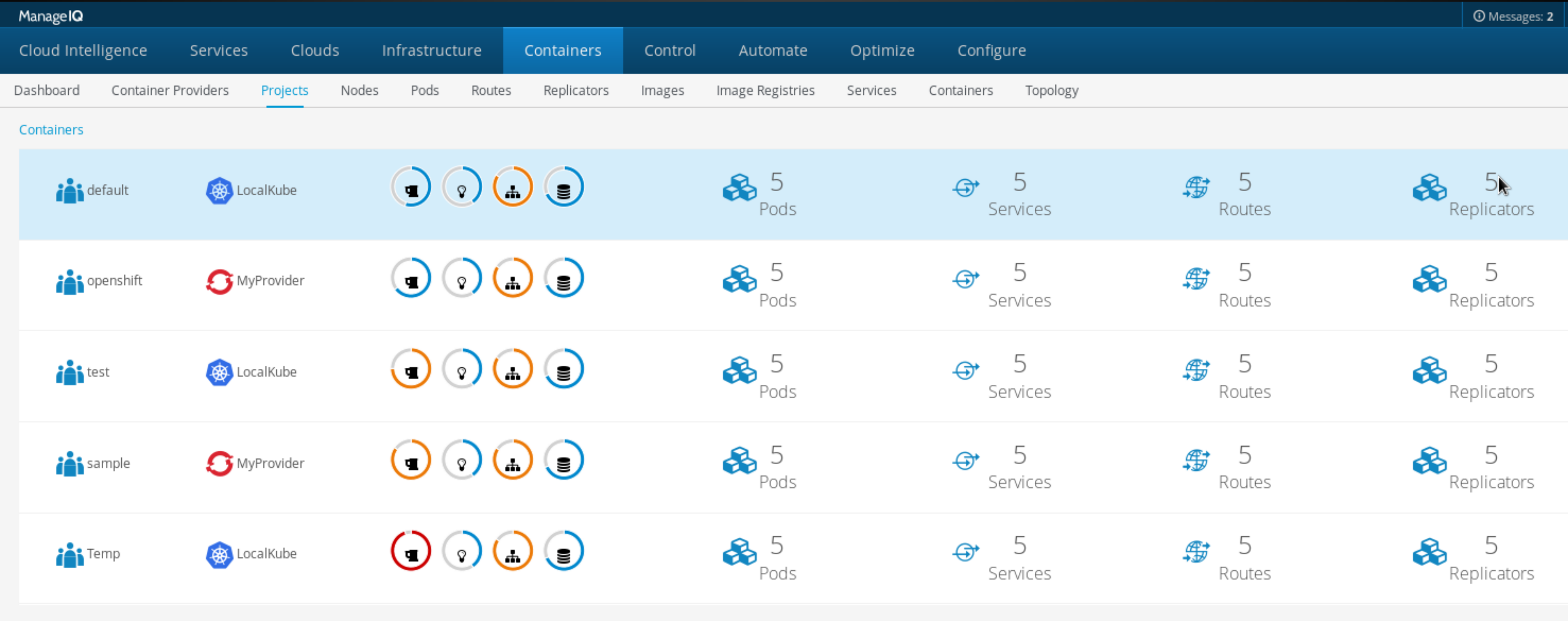
CPU

Memory

Total Pods Trends

Last 30 Days

Dashboard of a project



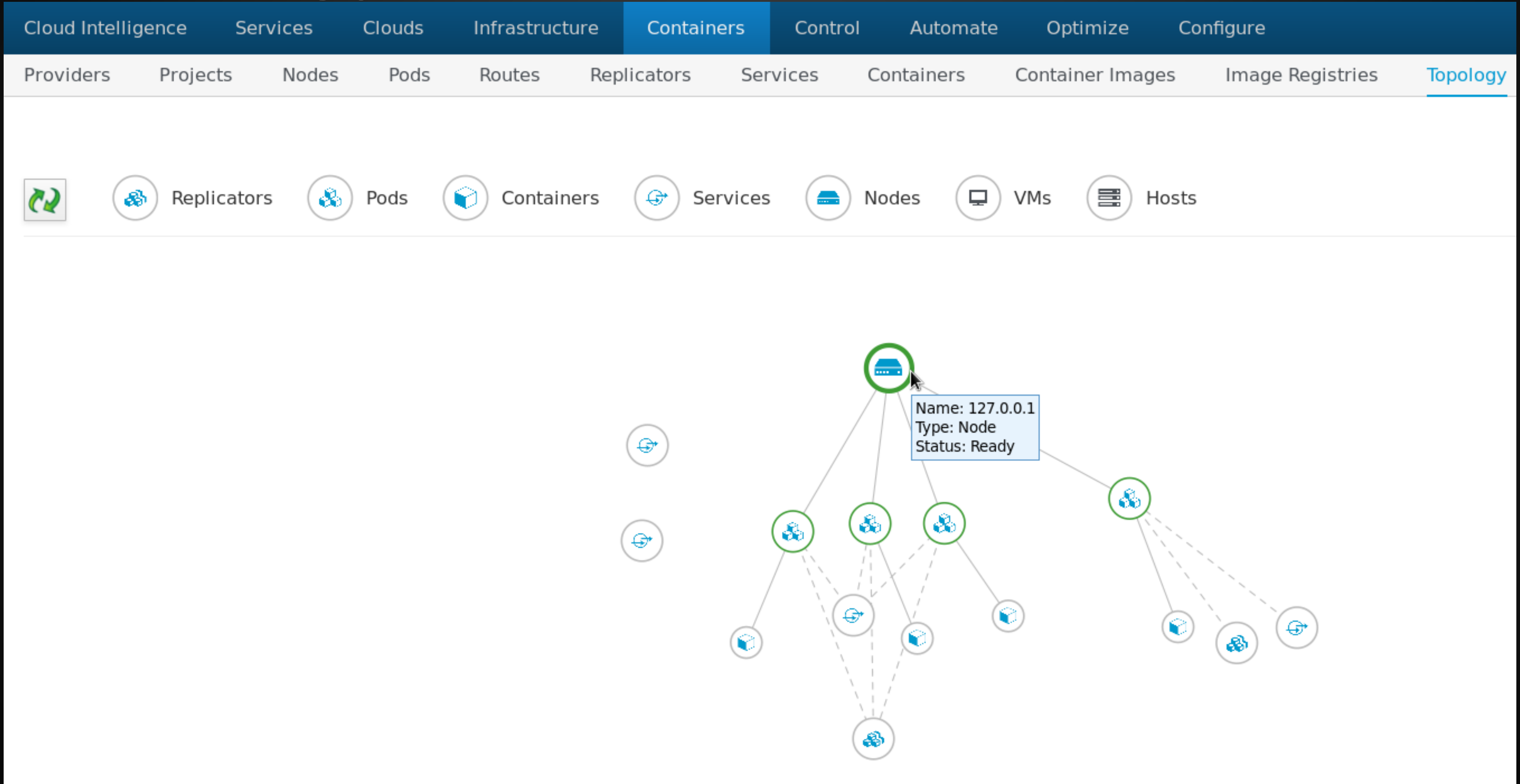
Containers

default	LocalKube	5 Pods	5 Services	5 Routes	5 Replicators
openshift	MyProvider	5 Pods	5 Services	5 Routes	5 Replicators
test	LocalKube	5 Pods	5 Services	5 Routes	5 Replicators
sample	MyProvider	5 Pods	5 Services	5 Routes	5 Replicators
Temp	LocalKube	5 Pods	5 Services	5 Routes	5 Replicators

Topology

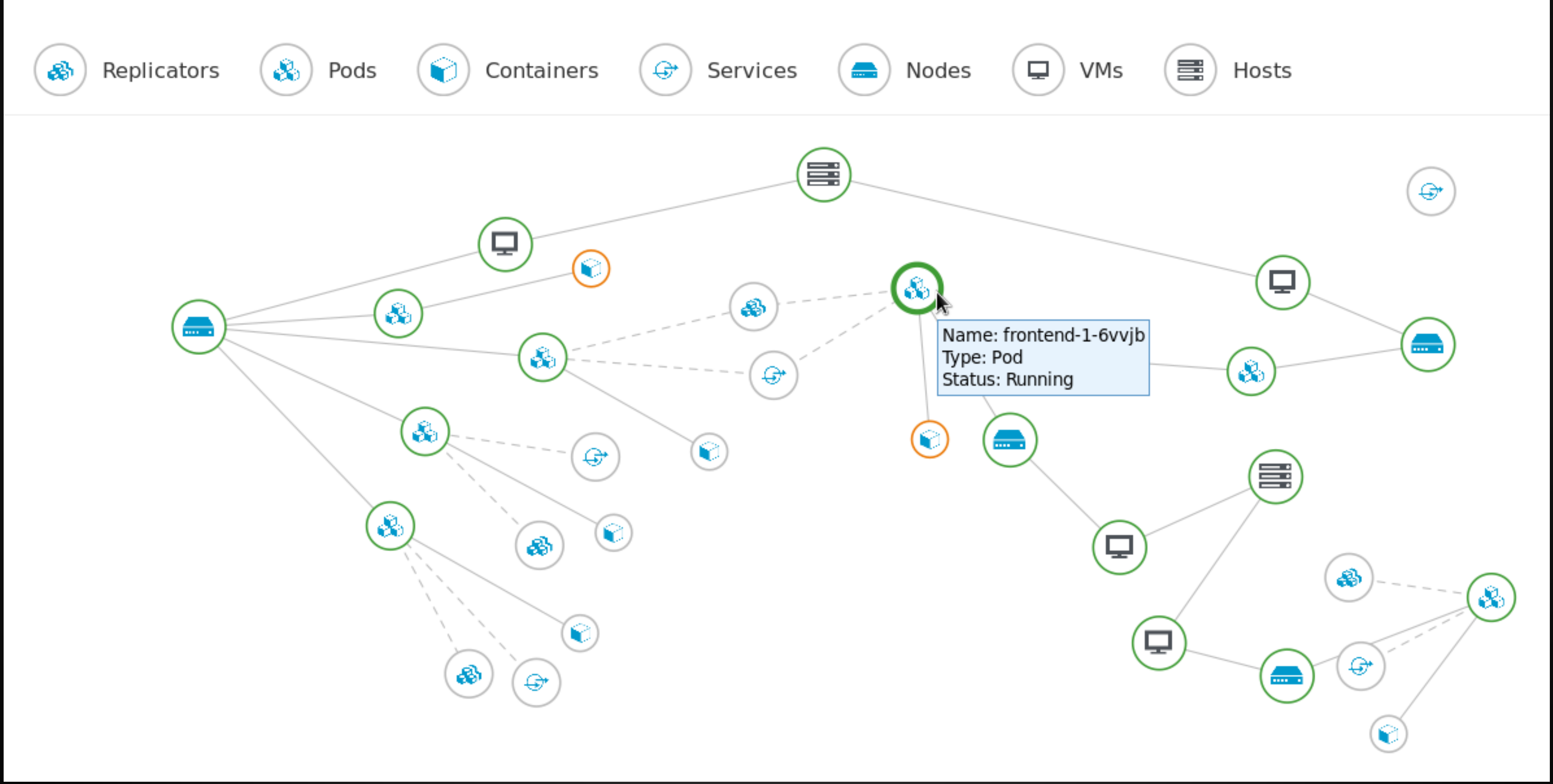
- “A picture is worth a thousand words”
- Includes cross provider relationships
- Statuses

Topology demo



Name: 127.0.0.1
Type: Node
Status: Ready

Topology demo



Name: frontend-1-6vvjb
Type: Pod
Status: Running

Open source

- Rapidly changing projects before their release
- Hard to track
- A chance to influence early

ManageIQ community

- http://talk.manageiq.org
- https://github.com/ManageIQ/manageiq
    - Follow label #providers/containers
- #manageiq IRC on freenode

Icons Credits

- The Go gopher - Renee French
- Diamond by MarkieAnn Packer from the Noun Project
- Rails by Luis Martins from the Noun Project
- Light Bulb by artworkbean from the Noun Project
- Sherlock by James Keuning

Thank you!

@mikeyteva