



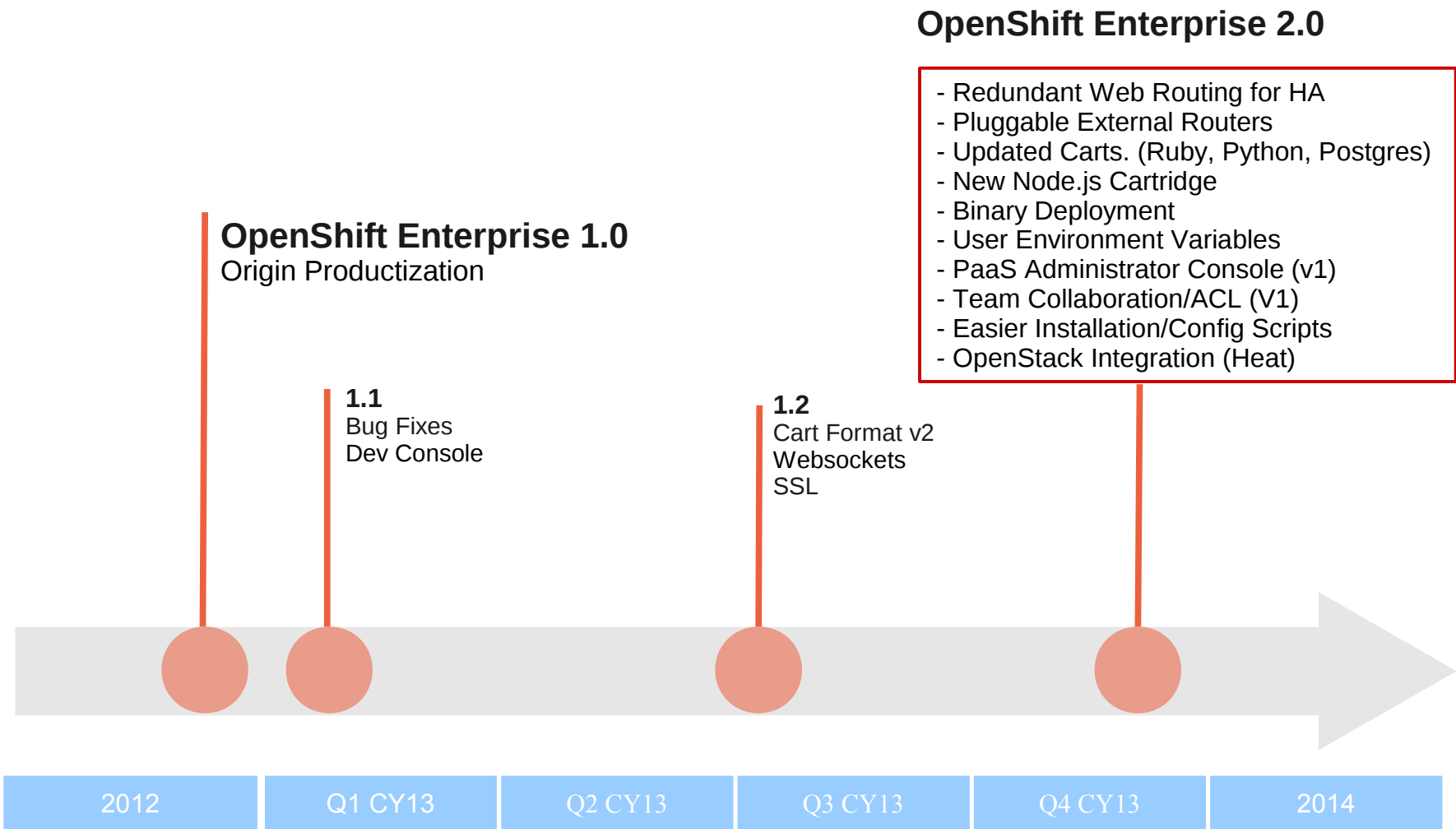
OpenShift Enterprise 2.0

Technical Overview

Ian Lawson

• **Senior Solution Architect, PubSec Tech Lead**

OpenShift Enterprise Release History



OpenShift Enterprise 2.0

- Redundant Web Routing for HA
- Pluggable External Routers
- Updated Carts. (Ruby, Python, Postgres)
- New Node.js Cartridge
- Binary Deployment
- User Environment Variables
- PaaS Administrator Console (v1)
- Team Collaboration/ACL (V1)
- Easier Installation/Config Scripts
- OpenStack Integration (Heat)

OpenShift Enterprise 1.0
Origin Productization

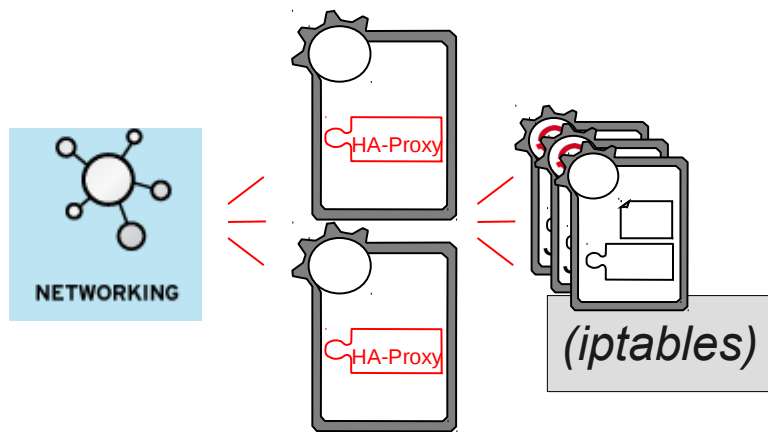
1.1
Bug Fixes
Dev Console

1.2
Cart Format v2
Websockets
SSL

2012 Q1 CY13 Q2 CY13 Q3 CY13 Q4 CY13 2014

Datacenter Routing Fabric

- ActiveMQ subscription method for pushing out key routing events
- Defined event type and string format
- Ruby Listener method for receiving events

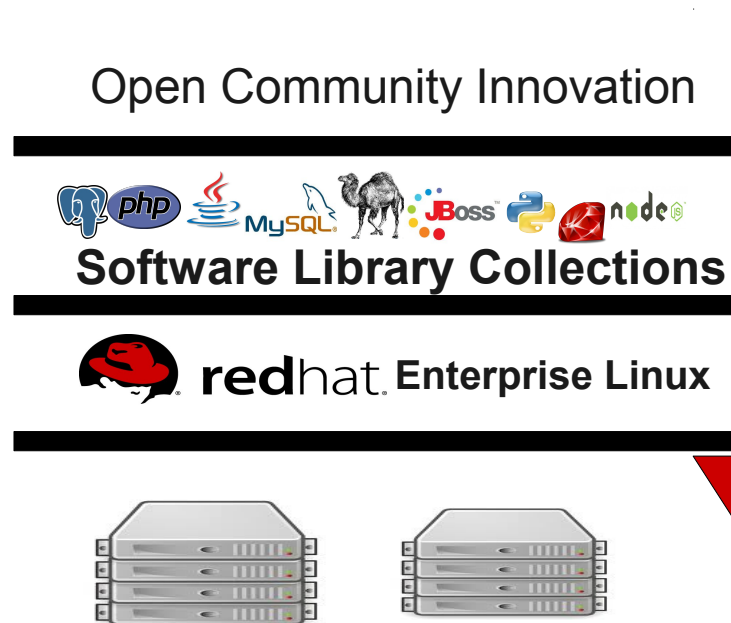


Event Type	String Format
app created	action => :create_application, :app_name => app.name, :namespace => app.domain.namespace
app deleted	action => :delete_application, :app_name => app.name, :namespace => app.domain.namespace
public endpoint created	action => :add_gear, :app_name => app.name, :namespace => app.domain.namespace, :public_port_name => endpoint_name, :public_address => public_ip, :public_port => public_port, :protocols => protocols, :types => types, :mappings => mappings
public endpoint deleted	action => :delete_gear, :app_name => app.name, :namespace => app.domain.namespace, :public_address => public_ip, :public_port => public_port
ssl cert added	action => :add_ssl, :app_name => app.name, :namespace => app.domain.namespace, :alias => fqdn, :ssl => ssl_cert, :private_key => pvt_key, :pass_phrase => passphrase
ssl cert removed	action => :remove_ssl, :app_name => app.name, :namespace => app.domain.namespace, :alias => fqdn
alias added	action => :add_alias, :app_name => app.name, :namespace => app.domain.namespace, :alias => alias_str
alias removed	action => :remove_alias, :app_name => app.name, :namespace => app.domain.namespace, :alias => alias_st

OSE 2.0 Polyglot Cartridges

- ◆ JBoss Enterprise Application Platform 6.1.0
- ◆ Jenkins Server
- ◆ Node.js 0.10
- ◆ Perl 5.10
- ◆ PHP 5.3
- ◆ Python 2.6
- ◆ Python 2.7
- ◆ Ruby 1.8
- ◆ Ruby 1.9
- ◆ Tomcat 6 (JBoss EWS 1.0)
- ◆ Tomcat 7 (JBoss EWS 2.0)
- ◆ Do-It-Yourself 0.1
- ◆ Cron 1.4
- ◆ Jenkins Client
- ◆ MySQL 5.1
- ◆ PostgreSQL 8.4
- ◆ PostgreSQL 9.2
- ◆ HA Proxy 1.4

Something Not Here?
(Supported Path)




Complete
Protection
RHN

OpenShift Enterprise

Cartridge Specification Guide

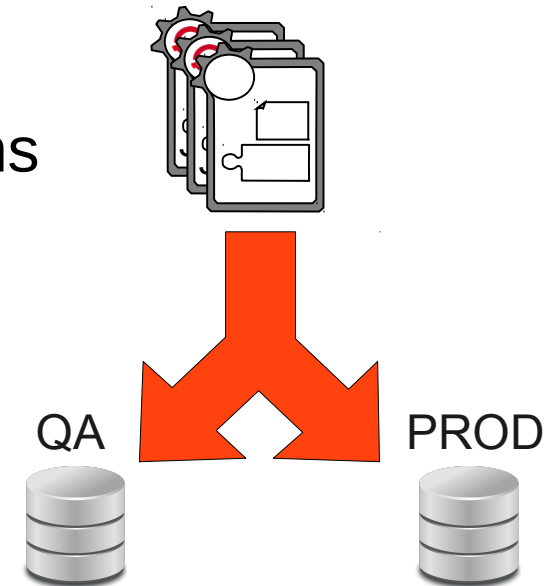
Specifications for developing OpenShift Enterprise cartridges



OPENSIFT
ENTERPRISE
by Red Hat

Work Versatility

- Set environmental variables at app creation and create/update/delete anytime thereafter
- Endless Possibilities to Single Cartridges
 - Modifying db passwords or source locations
 - Cartridge specific auth or license keys
 - App Clustering/Tokens
 - Changes to runtime process variables



```
$ rhc env list -a myapp
```

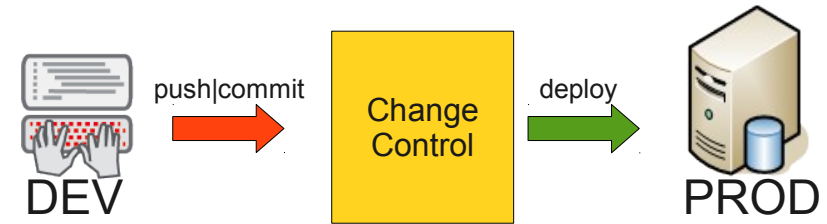
```
$ rhc env set Variable=Value Variable2=Value2 -a myapp
```

```
$ rhc env unset Variable -a myapp
```

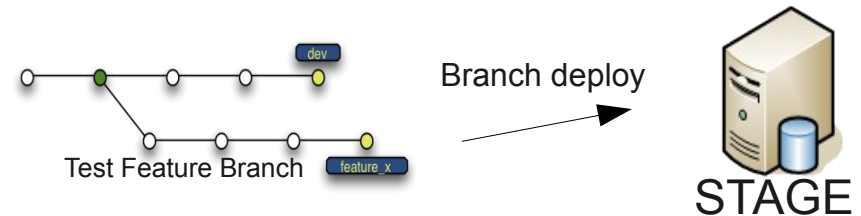
```
$ rhc env show Variable Variable2 -a myapp
```

Smart Deploy

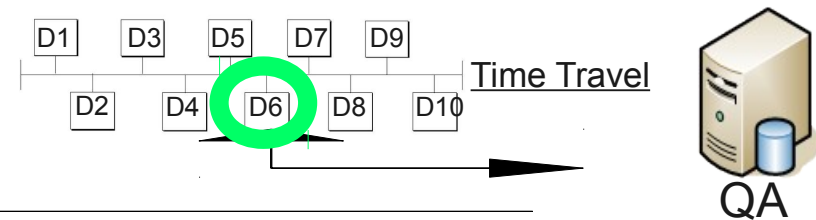
Control the behavior of a git push|commit but not deploy
`$ rhc configure-app -a app_name --no-auto-deploy`



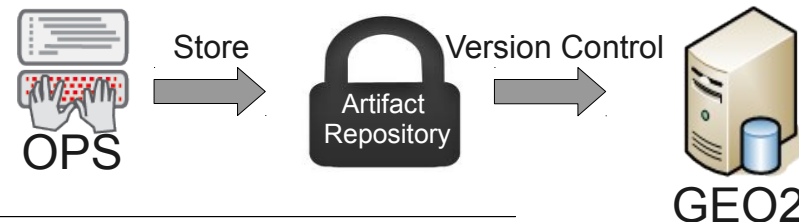
Deploy specific code branches, commit SHAs, or tags
`$ rhc deploy master -a app_name`



Turn on change recording and time travel to previous changes
`$ rhc configure-app -a app_name --keep-deployments 10`
`$ rhc deployments app_name`
`$ rhc activate-deployment deployment_ID -a app_name`



Decide if you are deploying code or binary
`$ rhc configure-app -a app_name --deployment-type binary|git`
`$ rhc deploy ./app.tar.gz -a app_name`
`$ rhc deploy http://foo.com/path/to/file.tar.gz -a app_name`



Scaled Apps Can have >1 HAProxy
When HAProxy < 1 : minimized downtime
When HAProxy > 1 : zero downtime

Graphical Visualization of Capacity

- Gear Profile Usage and Penetration Information
- District to Gear Consumption Information
- Node to Gear Usage Relationships
- Granular Table Summaries
 - Users, Gears, and Nodes
- Application Placement and Ownership Information
- Stats and Counts
- Usage Suggestions and Down Node Events

Highest Level: Gear Profiles

- Highest Level View Starts are Gear Profile Segmentation
 - Total gears are summarized by Districts and Nodes
 - Heat Map Characteristics
 - Darker the Color the more popular that Percentage
 - Orange = Threshold and red is over 100%

How Many of Each Profile

Small gears

DISTRICTS

24

GEARS

61,920

Total and Total Active

Total Capacity Numbers

MAX GEARS

144,000

How Many of Districts and Nodes

usage by district

100%

NODES

72

ACTIVE GEARS

5,961

Per District what are the total/max %

MAX ACTIVE GEARS

6,480

0%

usage by node

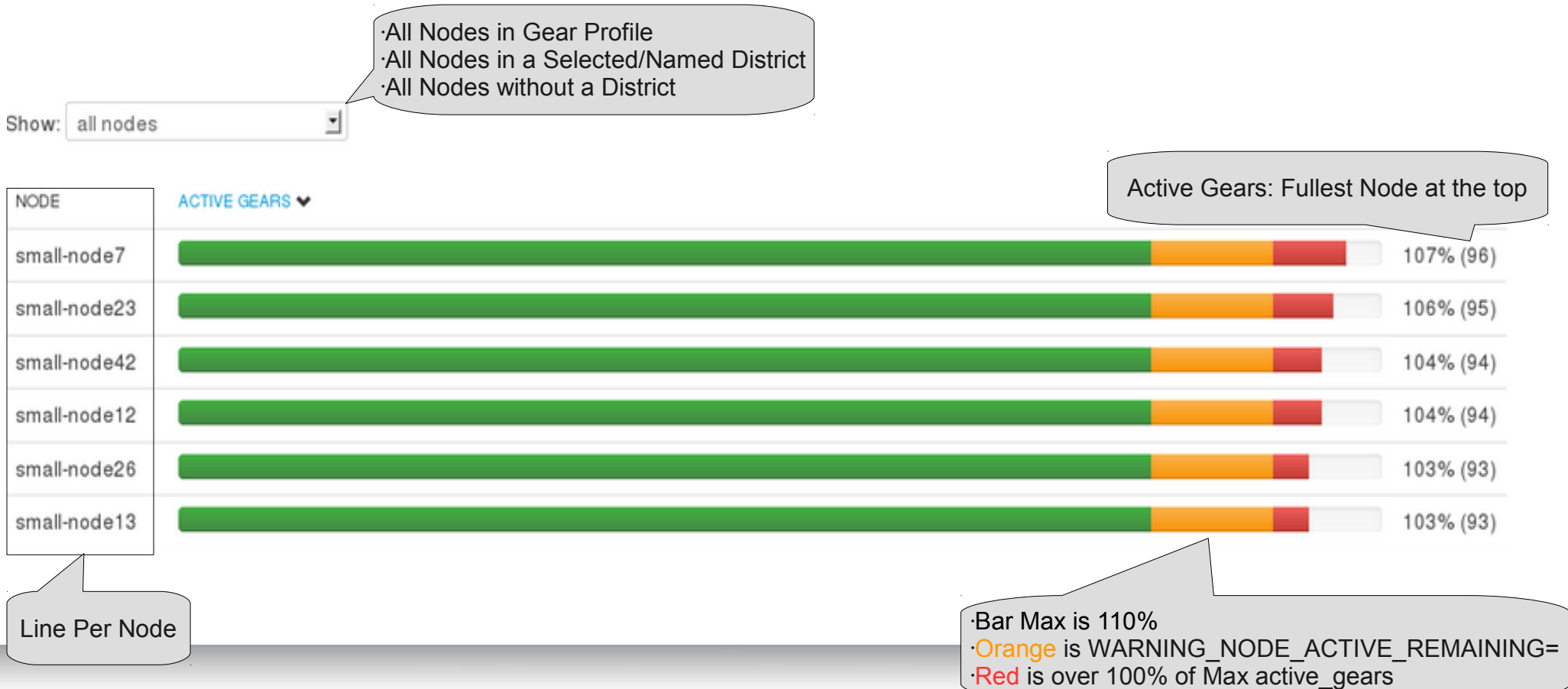
max_active_gears -

WARNING_NODE_ACTIVE_REMAINING

Per Node what are the active/max %

Nodes and Gears

- Drill Down into Nodes
 - Find Individual Node Consumption



Nodes and Gears: Summaries

- More Granular Information about Specific Nodes and Gears

Gear 527859f36892df1df3000...

 demo

Node [broker.example.com](#)
Application [wordpress-demo.example.com](#)
Cartridges PHP 5.3
MySQL 5.1

What cartridges are on the gear

What User is using the Gear

Node ip-10-77-9-118

Public Access

Hostname: [small-node03.example.com](#)
IP: 12.34.56.789

OS level Identity

Gears

Profile: small
Total: 6
Active: 6.67% (6 active of 90 max active)
Started: 6
Idle: 0
Stopped: 0
Deploying: 0
Unknown state: 0

Profile and total active %

Awesome Stats!

Applications

- Gears that make up an Application
- Cartridges that make up an Application
- Users and Domain namespace that own the Application

Application myapp

 janedoe

Users

URL: myapp-mydomain.dev.rhcloud.com

ID: 5231c103a750b1f154000005

Domain: mydomain

Good to know namespace

Gear Group for PHP 5.3, OpenShift Web Balancer, Jenkins Client, Cron 1.4

Gear	Cartridges	Node
5231c103a750b1f154000005	PHP 5.3, OpenShift Web Balancer, Jenkins Client, Cron 1.4	ip-10-77-9-118

Drill Down to Gear or Node

Gear Group for MySQL Database 5.1

Gear	Cartridges	Node
08f0ed001baf11e3abe512313d16fe8c	MySQL Database 5.1	ip-10-77-9-118

Discover how Apps are broken up across or within Gears

Suggestions

- Usage suggestions for Gear Profiles, Nodes, and Districts
- Threshold logic driven by variables set in the openshift-origin-admin-console.conf
 - GEAR_UP_THRESHOLD
 - GEAR_DOWN_THRESHOLD
 - GEAR_EXPECTED_ACTIVE_PERCENT
 - GEAR_UP_SIZE
 - WARNING_NODE_ACTIVE_REMAINING
- Not responding is a target list from mongoDB that is then mco pinged (needs to be in a District)

 4 nodes failed to respond
The nodes may be down, or may not have responded within the configured time

Server Identity

small-node5.example.com

small-node17.example.com

medium-node6.example.com

large-node51.example.com

From the broker, check if a node is still responding with:

```
mco ping -I <server_identity>
```

If the node is still responding then the configured timeout may be too short. Check

```
/etc/openshift/plugins.d/openshift-origin-admin-console.conf
```

If the node does not respond at all, the node may actually be unresponsive, or

While a node is not responding, its district is exempt from recommendations, because accurate district statistics cannot be calculated.

 Add capacity for small2 gear profile
Gear profile small2 has capacity for 100 active gears, which is below the threshold of 200.

Create 6 nodes with profile small2 to increase the active gear capacity. This suggestion is based on:

- 200 gears needed to both reach the threshold and meet the configured GEAR_UP_SIZE of this profile.
- The estimated maximum active gears of 50 for nodes in this profile.

Add 1 node to district district_2, which according to an active gear percent of 10% has room for 1 more node.

```
oo-admin-ctl-district -c add-node -n 'district_2' -i <server_identity>
```

Add 1 node to district district_3, which according to an active gear percent of 10% has room for 1 more node.

```
oo-admin-ctl-district -c add-node -n 'district_3' -i <server_identity>
```

Create 2 new districts.

```
oo-admin-ctl-district -c create -p 'small2' -n <name>
```

Then add 2 nodes to each new district.

```
oo-admin-ctl-district -c add-node -n '<district_name>' -i <server_identity>
```

Querying from Outside

- JSON Interface will change in future releases
- Way to extract information into different systems
 - Could also use oo-stats

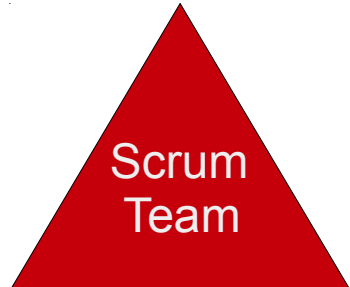
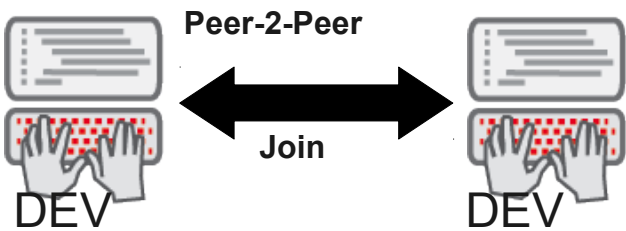
4.2. Exposed data

One of the goals for the admin console is to expose OpenShift system data for use by external tools. As a small step toward that goal, it is possible to retrieve the raw data from some of the application controllers as JSON. Note that this should not be considered the long-term API and is likely to change in future releases. You can access the following URLs when added to the appropriate server name, e.g. you could access `/admin-console/capacity/profiles.json` on the broker with:

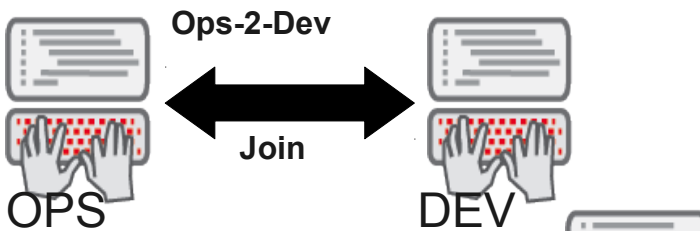
```
# curl http://localhost:8080/admin-console/capacity/profiles.json
```

- `/admin-console/capacity/profiles.json` - this returns all profile summaries from the Admin Stats library (the same library used by oo-stats). Add the `?reload=1` parameter to ensure the data is fresh rather than cached.
- `/admin-console/stats/gears_per_user.json` - this returns frequency data for gears owned by a user
- `/admin-console/stats/apps_per_domain.json` - this returns frequency data for apps belonging to a domain
- `/admin-console/stats/domains_per_user.json` - this returns frequency data for domains owned by a user

DevOps Memberships/Teams



High Velocity Iterations

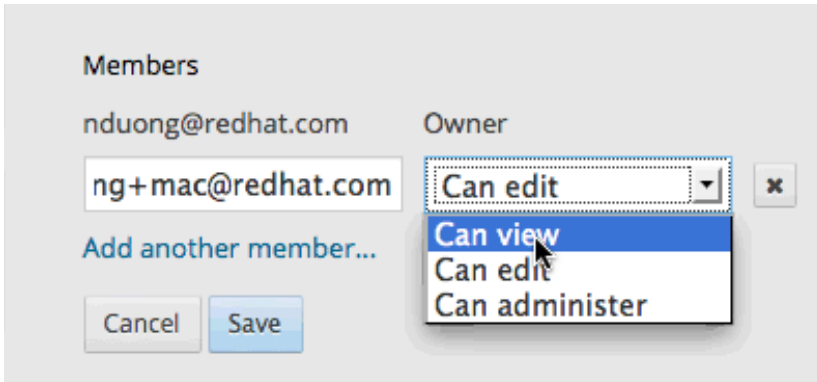


Life Cycle Relationships



Accessible via CMD, BUI, and IDE

```
$ rhc member add bob@example.com -n $DOMAIN_NAME -r edit  
$ rhc member list $DOMAIN_NAME  
$ rhc domain show  
$ rhc member remove -n $DOMAIN_NAME bob@example.com
```



JBoss Developer Studio 7.1 Integrated

User Roles

- App Owner assigns usernames and roles to the domain
 - **Viewer**: view application data (except for env variables)
 - **Editor**: edit/add/remove apps/cartridges/env vars and settings including git/ssh
 - **Admin**: adds modifying domain/team members plus roles (except for changing gear sizes or changing domain's name)
- Benefits
 - Add more people to memberships
 - Impose a Change Control Process



Enterprise Installation Enablement

- True Installer
 - Install environment remotely from a central server
 - Generation of configuration metadata for Forward Use
 - Puppet/Chef/Ansible
 - Offer a Clean Definition of Component Roles and End User Interrogation

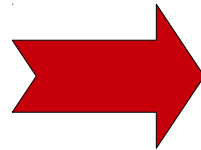
1. Install OpenShift Enterprise
 2. Add a Node to OpenShift Enterprise
- Type a selection and press <return>:

Host Information

Host	localhost
Roles	Broker, MsgServer, DBServer, Node
SSH Host	localhost
User	root
IP Addr	[unset]

Choose from the following deployment configuration options:

1. Change the DNS configuration
 2. Move an OpenShift role to a different host
 3. Modify the information for an existing host
 4. Add another Node host
 5. Finish editing the deployment configuration
- Type a selection and press <return>:



oo-install-cfg.yml

Description: This is the configuration file for the OpenShift Installer.
Version: 0.0.1
Vendor: OpenShift Origin Community
Subscription:
 type: none
Deployment:
DNS:
 app_domain: barrett.com
 register_components: "no"
Hosts:
- ip_addr: 192.168.1.25
 user: root
 ip_interface: eth0
roles:
- broker
- mqserver
- dbserver
- node
host: rhel-6-5-services
ssh_host: rhel-6-5-services

OpenStack HEAT Integration

- OpenShift [broker, MQ, mongoDB, node] Disk Images
- HEAT Orchestration Templates
 - To connect nodes to brokers

Images

The screenshot shows the OpenStack dashboard's 'Images & Snapshots' section. It lists three images:

- F18-x86_64-openshift-origin-node**: A disk image for OpenShift Origin nodes.
- F18-x86_64-openshift-origin-broker**: A disk image for OpenShift Origin brokers.
- F18-x86_64-cfnodds**: A disk image for OpenShift Origin nodes.

Each image entry includes a checkbox, the image name, and a 'Create Image' button. The interface also shows a 'Deploying 3 items' status and an 'Instance Snapshots' section below.

Result in OpenShift Deployed From OpenStack

The screenshot shows the OpenStack dashboard's 'Instances' section. It lists two instances:

Instance Name	IP Address	Size	Keypair	Status	Task	Power State	Actions
openshift.Nodeinstance	172.16.169.4	m1.medium 768MB RAM 1 VCPU 0 Disk	kraman_key	Active	None	Running	Create Snapshot
openshift.Brokerinstance	172.16.169.3	m1.medium 768MB RAM 1 VCPU 0 Disk	kraman_key	Active	None	Running	Create Snapshot

The interface also shows a 'Deploying 2 items' status and buttons for 'Launch Instance' and 'Terminate Instance'.

**Plus HEAT deployment
Metadata**

Case Studies

Business Challenge

- Provide a self-service PaaS to developers
- Reduce time to provision app platform from weeks/months to minutes
- Streamline pushing of code into production
- Reduce time to deploy new apps from 2-4 hours to <15 minutes

Why OpenShift?

- Streamline Developer enablement for Dev/Test
- Great efficiency for Prod deployments
- OpenShift on OpenStack

“Our motto is to Enable and Get Out of the Way. We need Self-service application stacks for developers.”

OPENSIFT
CASESTUDY



Business Challenge

- Large virtualization farm
 - 5000 Developers
 - 15,000 JVMs
- Want to add more automation, self service and improve productivity

Why OpenShift?

- Standardized Dev provisioning workflows
- Standardized stacks
- It just worked
- Technical depth and support from team

OPENSIFT
CASESTUDY

Business Challenge

- ISV with traditional on-premise offerings
- They are building a cloud offering too
- Want to improve their ability to innovate their cloud offering and manage their on-prem offerings
- See OpenShift as a platform to build on

Why OpenShift?

- Known Enterprise OS
- Feature-set
- Extensibility
- API Interfaces
- Ease of integration

OPENSIFT
CASESTUDY



Q&A