



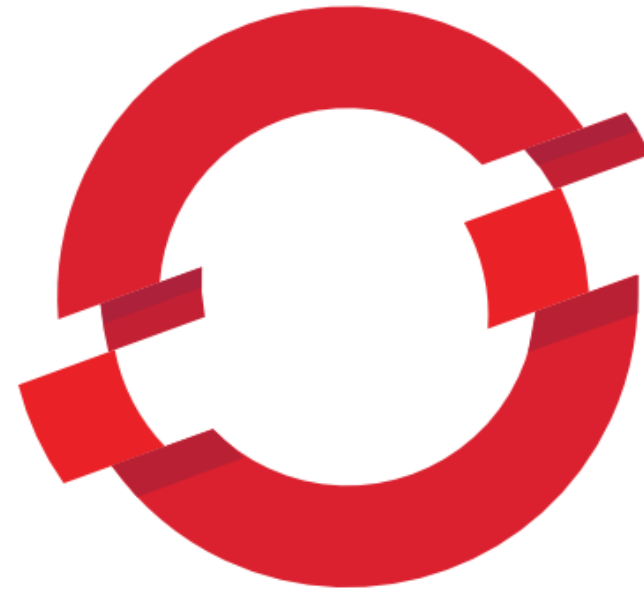
JUDCon

JBoss Users & Developers Conference

2013:India

Agenda

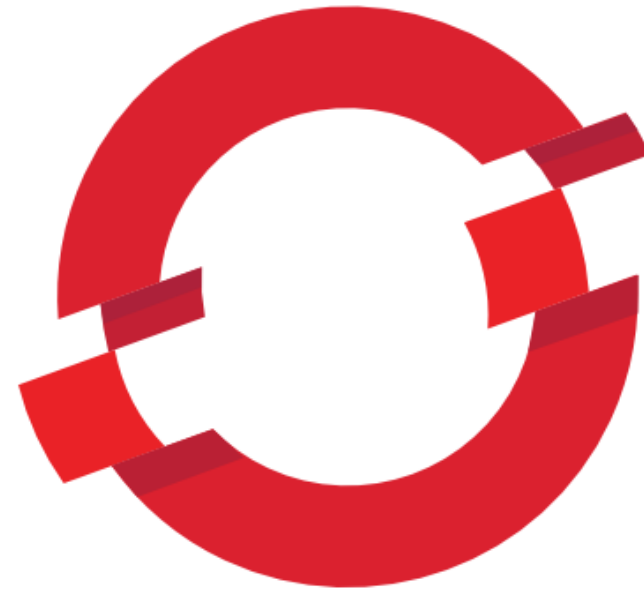
- Introduction - What is OpenShift?
- Architecture & Internals
- Cartridges
- Troubleshooting



OPENSSHIFT

Agenda

- Introduction - What is OpenShift?
- Architecture & Internals
- Cartridges
- Troubleshooting



OPENSSHIFT

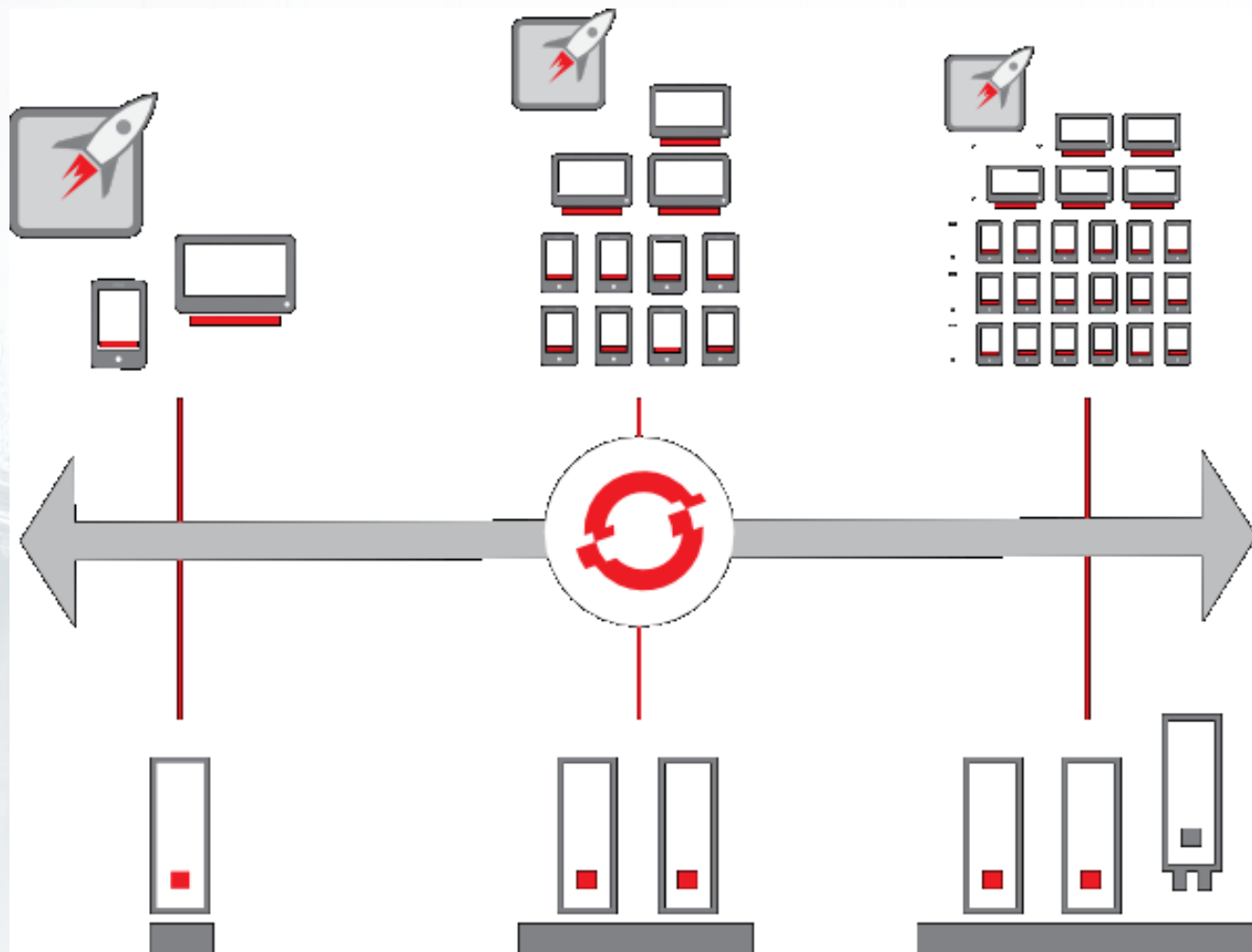
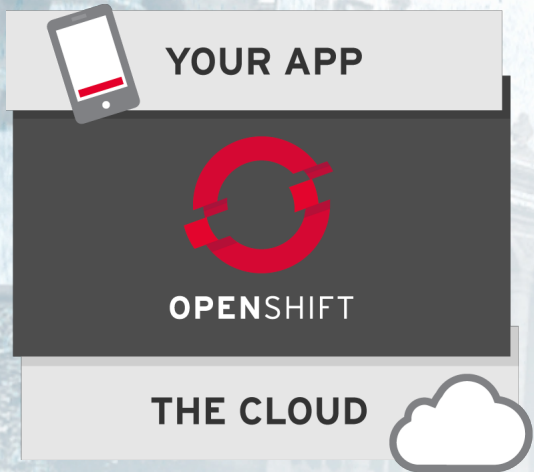
What's Platform As A Service?

Platform as a service (PaaS) is a category of cloud computing services that provide a computing platform and a solution stack as a service. Along with SaaS and IaaS, it is a service model of cloud computing. In this model, the consumer creates the software using tools and/or libraries from the provider. The consumer also controls software deployment and configuration settings. The provider provides the networks, servers, storage and other services.

PaaS offerings facilitate the deployment of applications without the cost and complexity of buying and managing the underlying hardware and software and provisioning hosting capabilities.

So Why OpenShift?

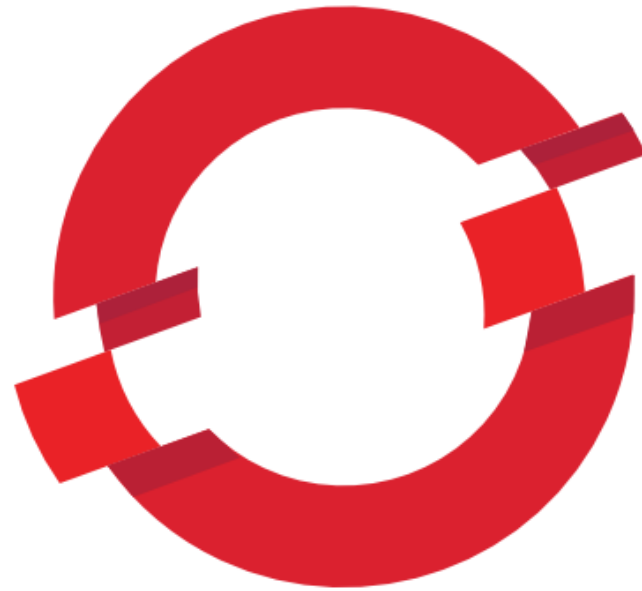
- Does it support multiple languages, frameworks and middleware?
- Does it provide support for multiple hosting environments?
- Is it extensible?
- Does it promote choice or lock-in?
- Does it auto-scale?



Short Demo !

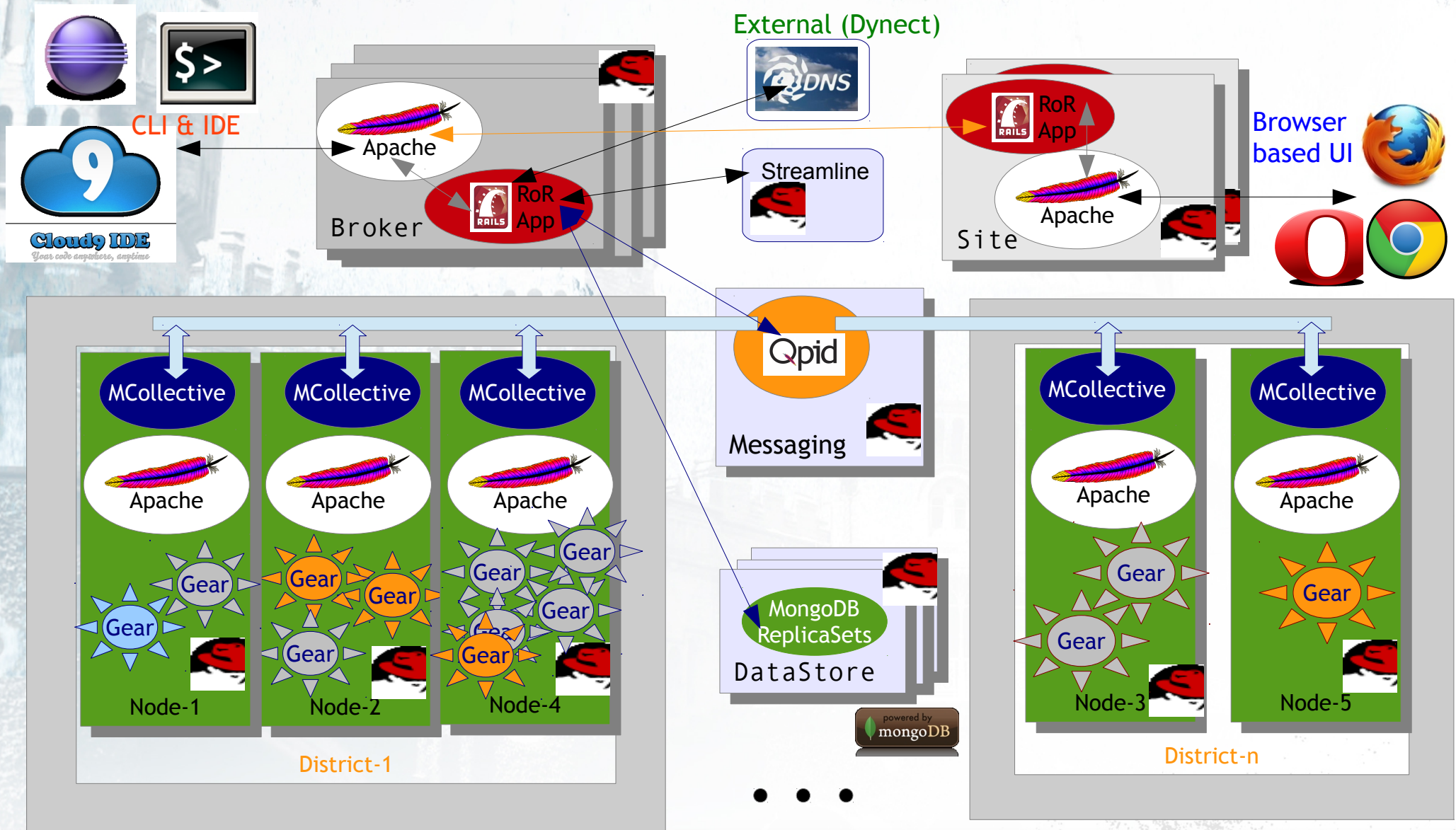
- `rhc domain create -n <namespace> [-l <login>]`
- `rhc app create -a <app_name> -t php-5.3 [-l <login>]`
- `cd <app_name>`
- `git remote add upstream -m master
git://github.com/openshift/phpmongotweet-example.git`
- `git pull -s recursive -X theirs upstream master`
- `git push # And that's to OpenShift.`

Architecture & Internals

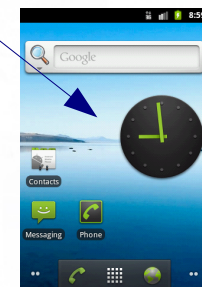
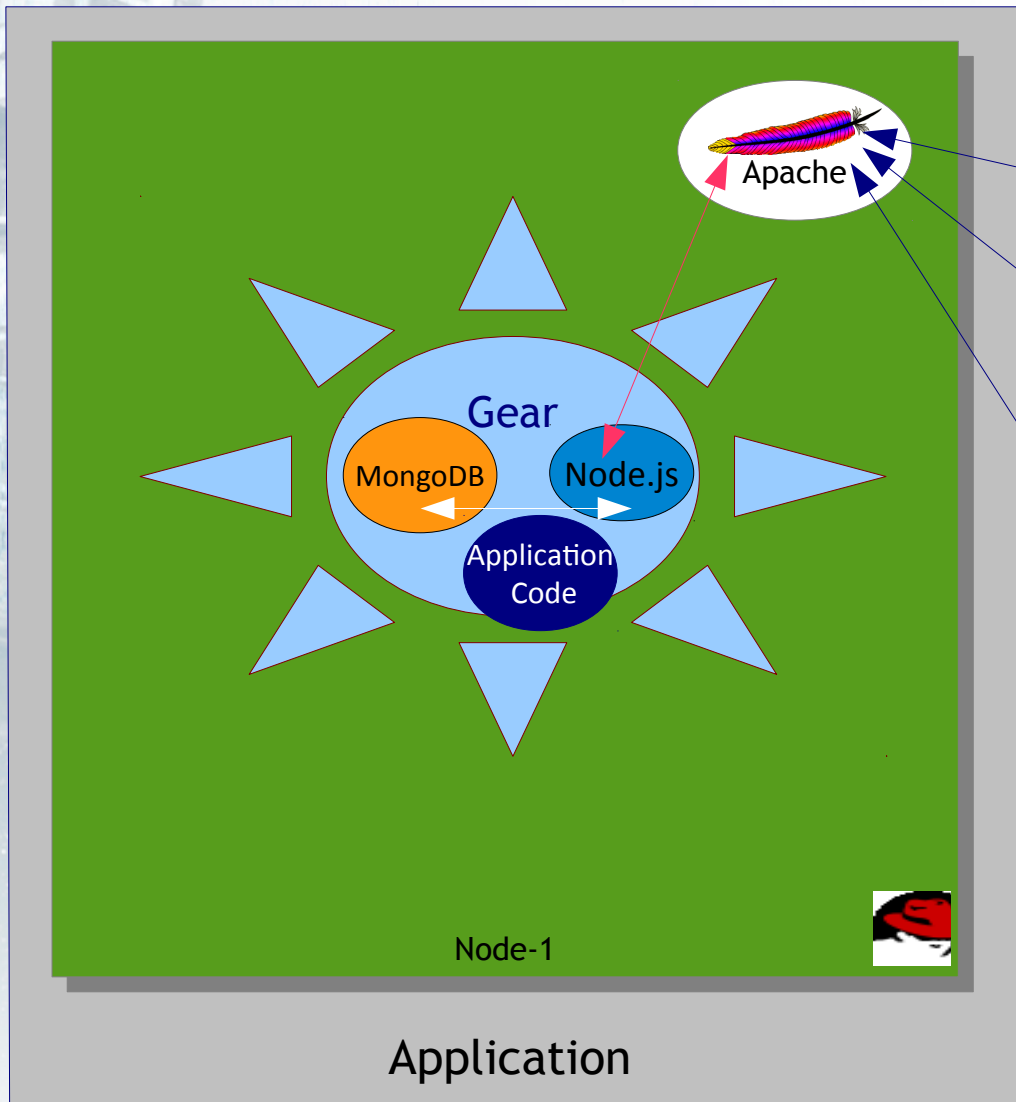


OPENSHIFT

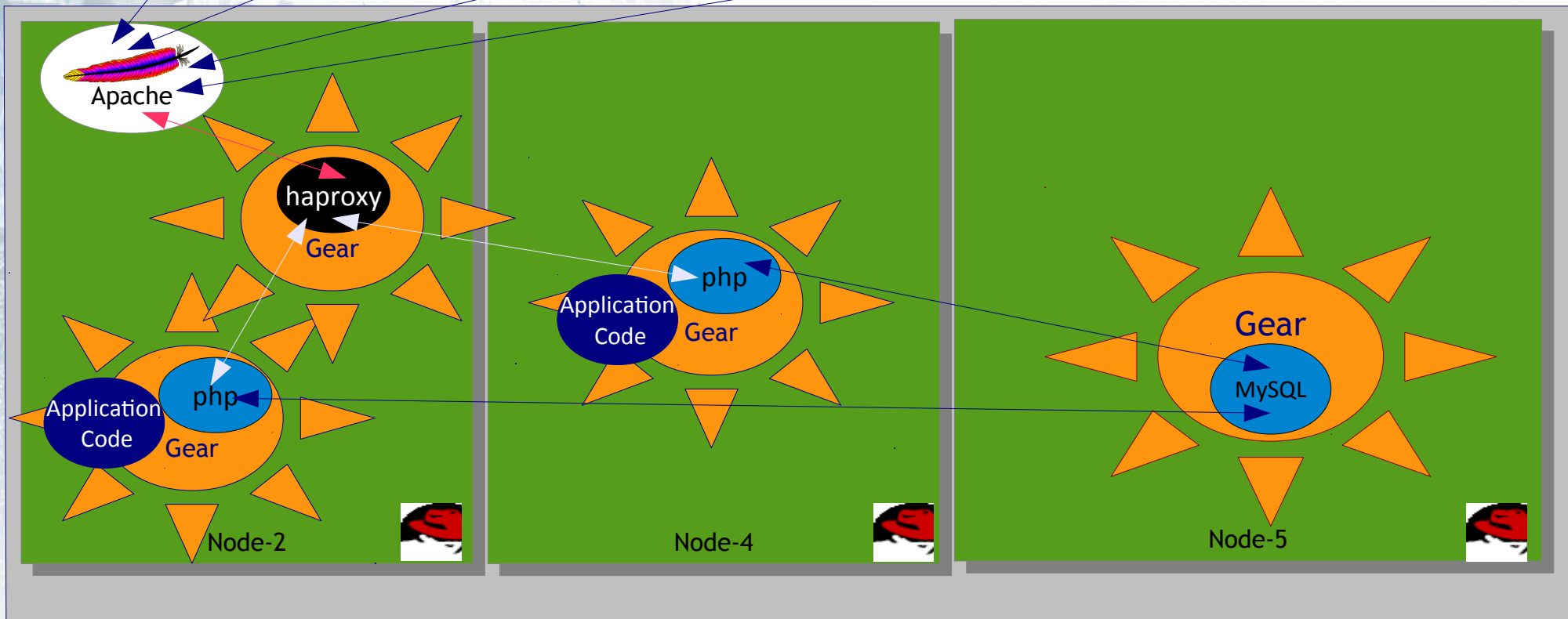
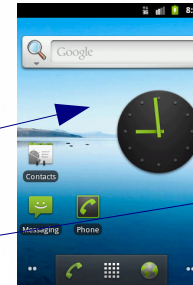
Okay, The Big Picture



“Tiny view” of a Application

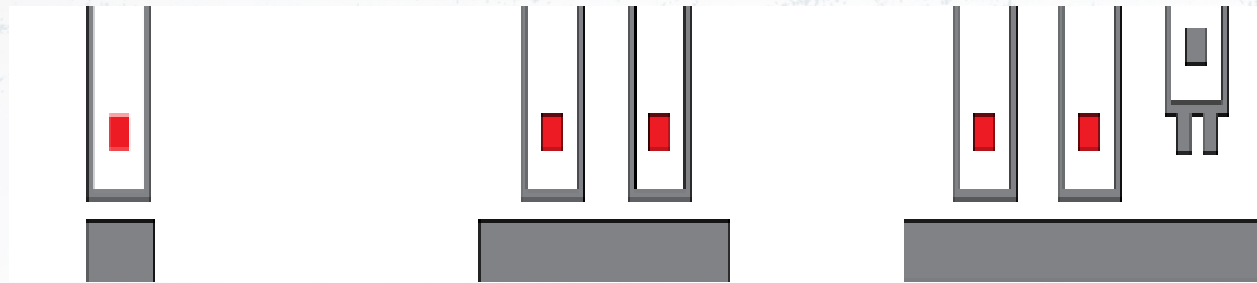


Anatomy of a scaled Application



Broker

is the single point of contact for all application management activities. It is responsible for managing user logins, DNS, application state, and general orchestration of the application. Customers don't contact the broker directly; instead they use the Web console, CLI tools, or JBoss tools to interact with Broker over a REST based API



To enable us to share resources, multiple gears run on a single physical or virtual machine. We refer to this machine as a node. Gears are generally over-allocated on nodes since not all applications are active at the same time.

ActiveMQ & MCollective

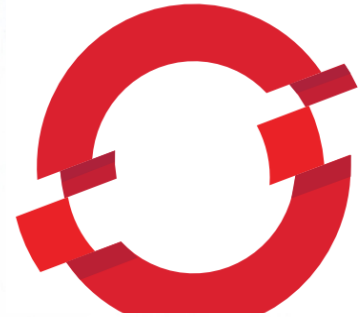
- ActiveMQ links the Brokers to the Nodes
- Nodes collection addressed via MCollective
 - Subscription to a QPID channel
 - Decentralization - the network is the only source of the truth
 - Broadcast - all servers get requests*
 - Node-Level filtering

Mongo DataStore



- User+namespace+application registry
- Production:
 - Triple-Modulo-Redundancy - 3 Replica Sets
 - With a couple of arbitrators [votes only]
 - Streamline does the user authentication
- Coming soon to a theatre near you
- Metering+Billing: Usage

Gears



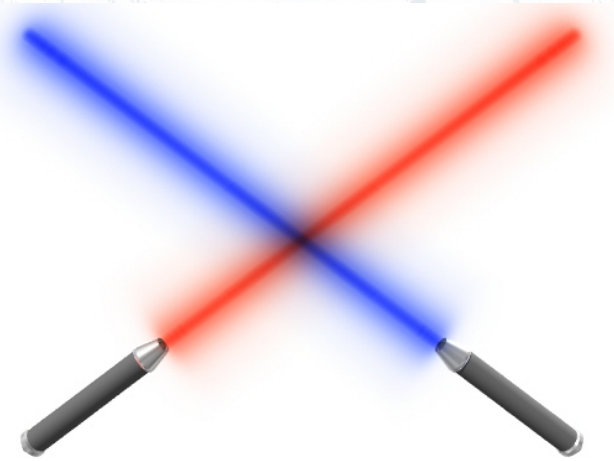
- Lightweight resource constrained container
- “Confined” linux user account
- Runs the application and/or associated services
- Flexible
- Limited amount of memory, cpu and storage
- CPU, Memory, IO [disk+network] constrained
 - via Linux Control Groups
- Bind allowed to allocated IP range %128 : 127.*.*.*

Gears ... (contd)

- Sand-boxed ...
 - Kernel Namespaces
 - Different View of the FileSystem
 - Bind Mounts
- Private Temps



Gears ... (contd)

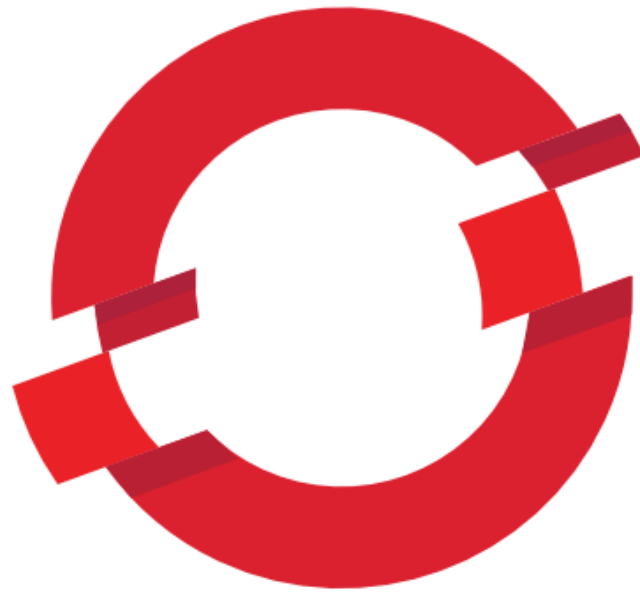


- Sand-boxed ...
- SELinux
- Unlearn “setenforce 0” && Permissive Mode
 - Use 'Z' force, Luke (&& /var/log/audit/audit.log)

```
[noder-rr95.dev.rhcloud.com repo]\> ls -altrZ server.js
-rwxrwxr-x. f6f79673053f41879e1eca087b3a3a06 f6f79673053f41879e1eca087b3a3a06
unconfined_u:object_r:libra_var_lib_t:s0:c0,c500 server.js

[noder-rr95.dev.rhcloud.com repo]\> ps -aefZ
LABEL                UID          PID  PPID  C  STIME TTY          TIME CMD
unconfined_u:system_r:libra_t:s0:c0,c500 500 23654 1   6 15:21 ?           00:00:00 node server.js
```


Internals



OPENSHIFT

Cartridges



- An extensible and stackable method of adding functionality to the underlying platform
- Huh??
 - Plugins
 - Run services multi-tenanted
 - Encapsulates best practices and recipes
- Provides a “standardized” method for management and control

Cartridges ... (contd)

- Frameworks/Languages:
 - JBoss/Java, PHP, Python, Ruby, Node.js, Perl
- Continuous Integration : Jenkins
- Databases: MySQL, PostgreSQL
- NoSQL: MongoDB
- Miscellaneous: CRON, phpMyAdmin, RockMongo
- Monitoring: 10gen MMS, Metrics*
- DIY* + Community Added* netty, Play, Python 2.7, Ruby 1.9 and more ...



Example Cartridge Manifests

- **php Manifest**

<https://github.com/openshift/crankcase/blob/master/cartridges/php-5.3/info/manifest.yml>

- **Node.js Manifest**

<https://github.com/openshift/crankcase/blob/master/cartridges/nodejs-0.6/info/manifest.yml>

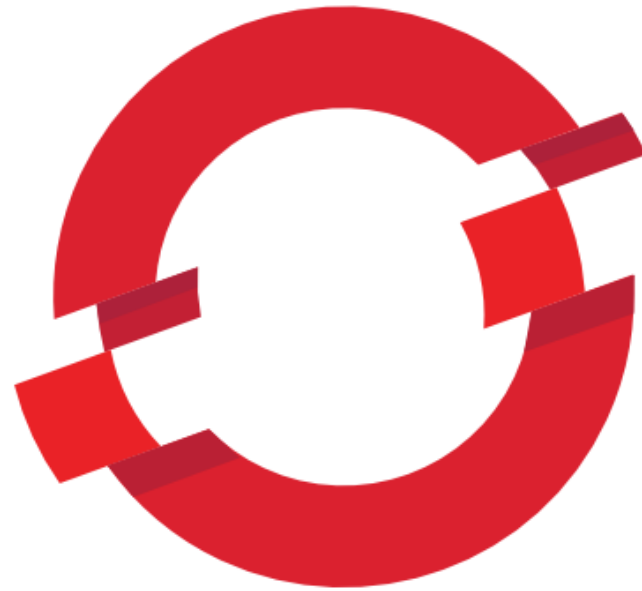
- **haproxy Manifest**

<https://github.com/openshift/crankcase/blob/master/cartridges/haproxy-1.4/info/manifest.yml>

- **MySQL Manifest**

<https://github.com/openshift/crankcase/blob/master/cartridges/mysql-5.1/info/manifest.yml>

OpenShift Origin



OPENSSHIFT

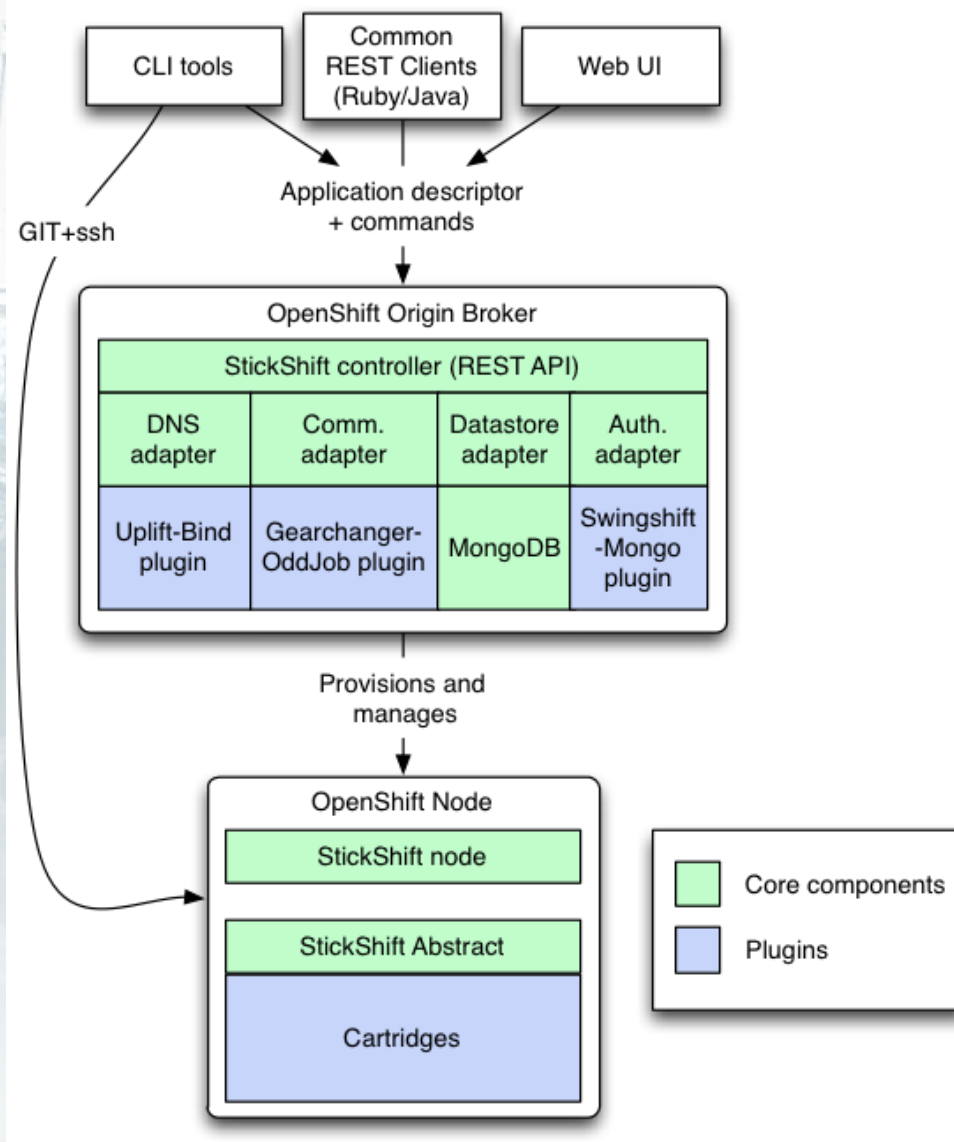
Take Me To The Source ...

- CrankCase
 - StickShift
 - UpLift
 - GearChanger
 - SwingShift
 - Cartridges



- <https://github.com/openshift/crankcase>

A Pictorial View



“The Closed Bits”



- Not all bits open-sourced
- RH Specific due to containing internal information + some security policies
 - Dynect Plugin: Internal interface to Dynect
 - Streamline Plugin: RedHat specific
 - SELinux policies: Multi-tenancy lockdowns
- MCollective Plugin: On its way
- Site: UI [still has some RH specifics] && !docs

Cartridges



OPENSSHIFT

Cartridges



- An extensible and stackable method of adding functionality to the underlying platform
- Huh??
 - Plugins
 - Run services multi-tenanted
 - Encapsulates best practices and recipes
- Provides a “standardized” method for management and control

Cartridges Contents

- Hooks
- Connectors (Connection-Hooks)
- Manifest
- Helper Scripts
- Templates

[crankcase](#) / [cartridges](#) / [haproxy-1.4](#) / [info](#)

name	age	message	history
..			
bin	2 days ago	Fix for bugz 820051 - Allow scalable gears to be moved. [Ram Ranganathan]	
build	23 days ago	initial commit [danmcp]	
configuration	23 days ago	initial commit [danmcp]	
connection-hooks	23 days ago	initial commit [danmcp]	
data	23 days ago	initial commit [danmcp]	
hooks	2 days ago	Fix for bugz 820051 - Allow scalable gears to be moved. [Ram Ranganathan]	
changelog	23 days ago	initial commit [danmcp]	
control	23 days ago	initial commit [danmcp]	
manifest.yml	23 days ago	initial commit [danmcp]	

[crankcase](#) / [cartridges](#) / [php-5.3](#) / [template](#)

name	age	message
..		
.openshift	23 days ago	initial co
libs	23 days ago	initial co
misc	23 days ago	initial commit [danmcp]
php	23 days ago	initial commit [danmcp]
README	6 days ago	Bug 819739 [danmcp]
deplist.txt	23 days ago	initial commit [danmcp]

Cartridge Hooks

- Configuration
 - [pre-,post-] {install | remove}
 - [pre-,post-] {configure | deconfigure}
 - {add | remove} - alias
 - {add | remove} - module
 - {deploy | remove} - httpd-proxy
 - update-namespace



Cartridge Hooks ... (contd)



- Lifecycle
 - [pre-,post-] move
- [pre-,post-] {start | stop | force-stop | restart | reload}
 - status
- Informational - info
- Management
 - {show | conceal | expose} - port
 - system-messages | threaddump | tidy

Cartridge Connection-Hooks

- Ad-Hoc
- Between Publishers and Subscribers
 - Connected based on the type
 - FileSystem
 - FileSystem:Shared
 - NET:Unix
 - NET:TCP:Internal
 - NET:TCP



Example Cartridge Connectors

- **php Manifest**

<https://github.com/openshift/crankcase/blob/master/cartridges/php-5.3/info/manifest.yml>

- **Node.js Manifest**

<https://github.com/openshift/crankcase/blob/master/cartridges/nodejs-0.6/info/manifest.yml>

- **haproxy Manifest**

<https://github.com/openshift/crankcase/blob/master/cartridges/haproxy-1.4/info/manifest.yml>

- **MySQL Manifest**

<https://github.com/openshift/crankcase/blob/master/cartridges/mysql-5.1/info/manifest.yml>

Cartridges

- By Example:
 - **Node.js** : <https://github.com/openshift/crankcase/tree/master/cartridges/nodejs-0.6>
 - **MongoDB**: <https://github.com/openshift/crankcase/tree/master/cartridges/mongodb-2.0>
- Application using Node.js and MongoDB
 - <https://github.com/ramr/pacman>