

*A tour of*



*from the*

# COMMAND LINE

*featuring:*

*ssh, git, and rhc*

[bit.ly/1mq7s5h](http://bit.ly/1mq7s5h)

[socuteurl.com/widdlepuppytail](http://socuteurl.com/widdlepuppytail)



presented by

[ryan jarvinen](#) / [@ryanj](#)



# Open Source Evangelist at Red Hat

[ryanj@redhat.com](mailto:ryanj@redhat.com)



# AGENDA

1. [State of the Open Cloud](#)
2. [rhc](#)
3. [ssh](#)
4. [git](#)
5. [Release Management](#)
6. [Scaling](#)



# STATE OF THE OPEN CLOUD



# HAS IT BEEN LIBERATED?

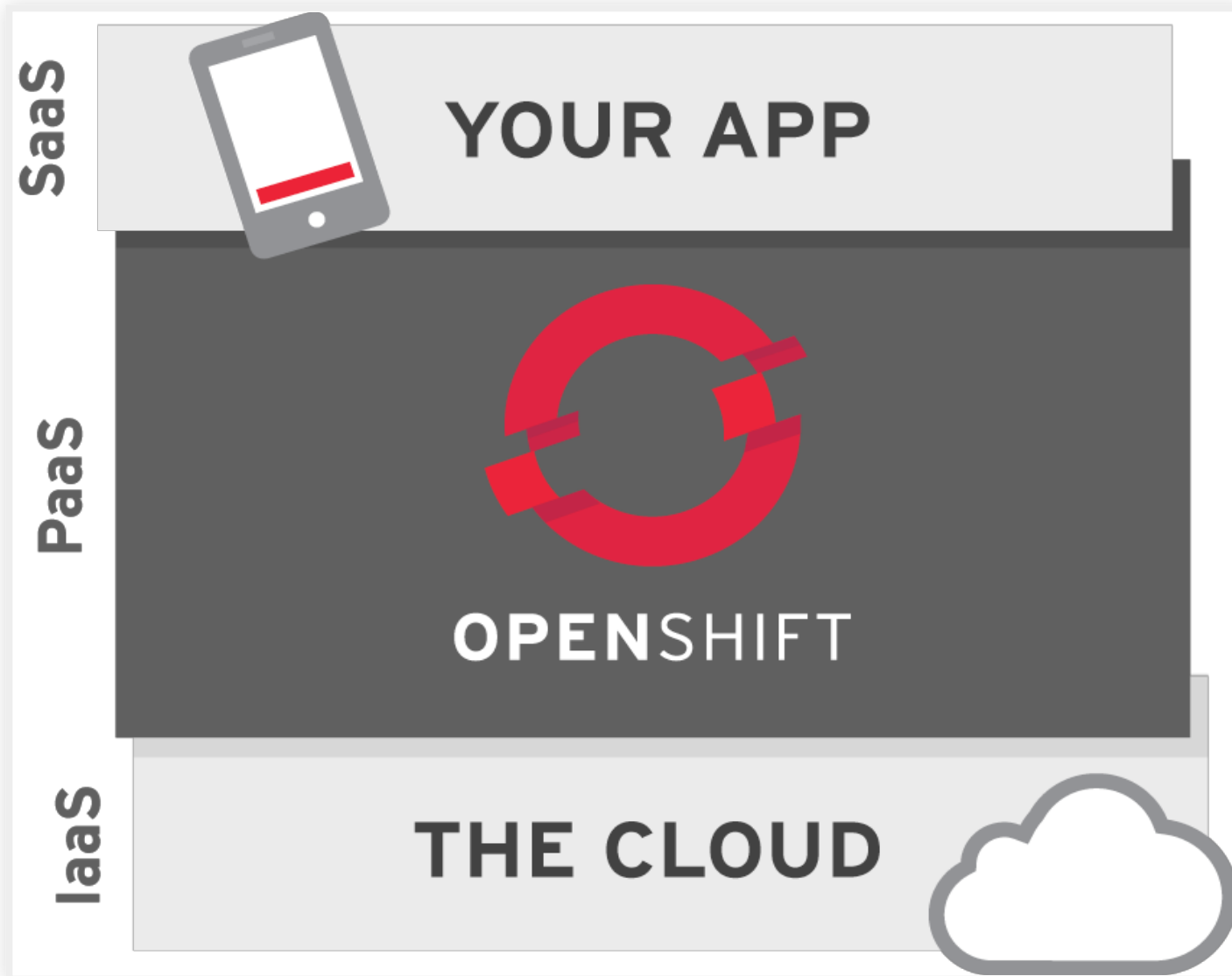
1. The OS ✓
2. The Cloud ?
3. Infrastructure as a service ✓
4. Platform as a service ✓
5. Software as a service ×



# CLOUD AUTOPSY







**OPENSIFT IS...**

**OPEN SOURCE**

**HOSTING,**

**BUILD,**

*and*

**AUTO-SCALING**

for applications



providing instant access to all of this, and more:



**DEVELOPER TOOLS**  
eclipse, Built by: maven, Jenkins, appcelerator

**PACKAGED APPS**  
WordPress, Joomla!, Drupal, SUGARCRM, JASPER SOFT

**FRAMEWORKS**  
Sinatra, Bottle, Seam, G, RAILS, spring, django, symfony, ZF ZEND FRAMEWORK

**LANGUAGES**  
python, Java, php, node, Ruby, perl

**CUSTOM CARTRIDGES**  
Play!, Couchbase (Membase), M, [dashed arrow]

**MIDDLEWARE**  
Apache Tomcat, APACHE, JBoss by Red Hat

**DATASTORES & BACKENDS**  
PostgreSQL, mongoDB, MySQL

**HOSTING**  
amazon web services

**OS**  
Red Hat Enterprise Linux 6

**OPENSIFT**



these technologies are bundled / made available as  
**CARTRIDGES**





```
rhc cartridge list
```



On OpenShift, your application environment is securely incapsulated within a '*Gear*', providing guaranteed access to system resources



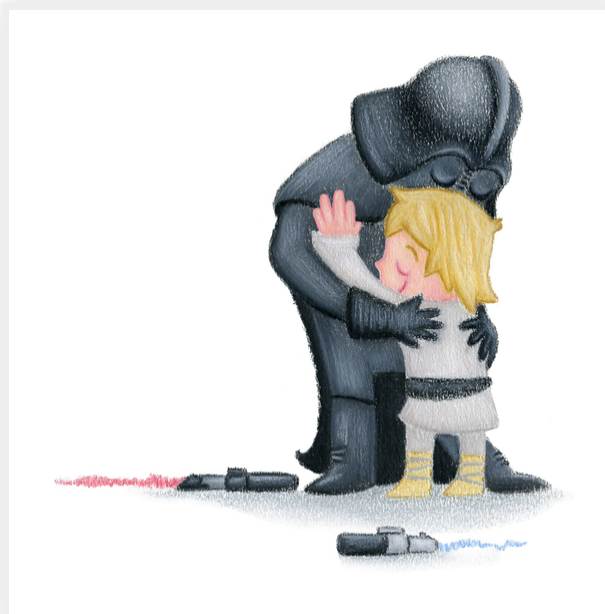


using SELinux, and Cgroups





# OPENSIFT PROVIDES A PEACEFUL ENVIRONMENT FOR DEVS AND SYSADMINS TO WORK TOGETHER IN



- **Operations care about stability and performance**
- **Developers just want environments without waiting**
- **And neither one wants to have to fight the other to get their work done**



# RHC



# rhc makes it easy for developers to talk to your OpenShift REST API

## **PRE-REQUISITES:**

rubygems, git

```
sudo gem install rhc
```



```
rhc setup
```

Will automatically:

- authenticate your OpenShift account
- verify your local ssh key configuration
- configure your host url identifier



# CREATE AN APPLICATION

Start by provisioning your application environment and database in a single step:

```
rhc app create APP_NAME CARTRIDGE CART2 CART3
```

for a basic Nodejs and MongoDB application, run:

```
rhc app create scale12x nodejs-0.10 mongodb-2.2
```



## <command-line output>

### Application Options

-----

```
Namespace:  shifter
Cartridges: nodejs-0.10, mongodb-2.2
Gear Size:  default
Scaling:    no
```

Creating application 'scale12x' ... done

Waiting for your DNS name to be available ... done

Downloading the application Git repository ...

Cloning into 'scale12x'...

Your application code is now in 'scale12x'



```
scale12x @ http://scale12x-shifter.rhcloud.com/ (uuid: 5175981a59
-----
Created:          1:05 PM
Gears:           1 (defaults to small)
Git URL:
ssh://5175981a5973ca7a69000501@scale12x-shifter.rhcloud.com/~/.git
SSH:            5175981a5973ca7a69000501@scale12x-shifter.rhcloud.com

nodejs-0.10 (Node.js 0.10)
-----
Gears: Located with mongodb-2.2

mongodb-2.2 (MongoDB NoSQL Database 2.2)
-----
Gears:          Located with nodejs-0.10
Connection URL:
mongodb://$OPENSIFT_MONGODB_DB_HOST:$OPENSIFT_MONGODB_DB_PORT/
Database Name:  scale12x
Password:      PTk4cCetTj2w
Username:      admin

RESULT:
Application scale12x was created.
```



redhat



# SUCCESS!

You now have a basic node.js and MongoDB skeleton application live on the Internet!

<http://scale12x-shifter.rhcloud.com/>

Your gear is now configured with:

- it's own git repo
- it's own web server
- ssh access
- logging
- a database



● publicly accessible hostnames, automatic DNS

Application details are always available via:

```
rhc app show scale12x
```

You can tail your remote logs with:

```
rhc tail scale12x
```

or, connect directly to your app via ssh:

```
rhc ssh scale12x
```



# SSH



# ENVIRONMENT VARIABLES

Allow you to write code that will run anywhere

```
//provide a sensible default for local development
mongodb_connection_string = 'mongodb://127.0.0.1:27017/' + db_name
//take advantage of openshift env vars when available:
if(process.env.OPENSIFT_MONGODB_DB_URL){
  mongodb_connection_string = process.env.OPENSIFT_MONGODB_DB_URL
}
```

```
//same advice applies for your webserver's PORT and IP address
var port = process.env.PORT || process.env.OPENSIFT_NODEJS_PORT
var ip    = process.env.OPENSIFT_NODEJS_IP || '127.0.0.1'
```



## Application passwords, keys, and secrets can be abstracted using the same technique:

```
rhc env set SECRET_KEY=0P3N_S0URC3
```

```
rhc env list
```

```
rhc env help
```



## Team members can supply their own keys during app creation, for a single step clone+deploy:

```
rhc app create scale12x nodejs-0.10 mongodb-2.2 \  
--from-code=http://github.com/USER/TEAM_REPO.git \  
--env SECRET_KEY=0P3N_S0URC3
```



# TEAM COLLABORATION

There are several ways to collaborate:

1. using [github or bitbucket](#)
2. using [ssh keys](#)
3. or, using [OpenShift's new team collaboration tools](#)



# GIT





# DEPLOYING UPDATES

A standard git development workflow can be used to rebuild and update your remote application:

## 1. Add your changes to a changeset

```
git add index.html
```

## 2. Mark the changeset as a Commit

```
git commit -m 'updating H1'
```

## 3. Push the Commit to OpenShift

```
git push
```



## Adding cartridges to existing apps is easy:

```
rhc cartridge add jenkins-1
```

adds jenkins CI to your application's build cycle



# RELEASE MANAGEMENT



## RELEASE TRACKING & ROLLBACKS

```
rhc deployment show
```

```
rhc deployment list
```

```
rhc deployment activate CHECKSUM
```

Want to deploy a different branch (not 'master')?

```
rhc app configure --deployment-branch MY_BRANCH
```

<https://www.openshift.com/blogs/introduction-to-deployments-and-rollbacks-on-openshift>



# TIPS FOR LOCAL DEVELOPMENT

Use port-forwarding to create a local connection to your remote database instance:

```
rhc port-forward scale12
```

Starting a local webserver is different in each language. For nodejs, you can start a local server with:

```
npm install
```

```
npm start
```

<https://www.openshift.com/blogs/set-up-local-access-to-openshift-hosted-services-with-port-forwarding>



# SCALING



## Spinning up an auto-scaling Linux environment:

```
rhc app create scale12 -s nodejs-0.10 mongodb-2.2
```

just add "-s" to your app create command



## Set a min and max scale

```
rhc cartridge scale nodejs-0.10 -a scale12 --min 2 --max 12
```

## or, manually scale an application

```
rhc app scale-up
```

```
rhc app scale-down
```





## From inside a hosted environment:

```
haproxy_ctld --up
```

```
haproxy_ctld --down
```

## HAProxy on / off switches

```
haproxy_ctld_daemon start
```

```
haproxy_ctld_daemon stop
```

```
haproxy_ctld_daemon restart
```



# GENERATING LOAD

For scalable applications, generate load from the command line to see automatic scaling in action:

```
while true ; do ab -n 1000 -c 50 https://scale12-shifter.rhcloud.
```



# HAPROXY WEB UI

[http://\\$YOUR\\_APP\\_DNS/haproxy-status/](http://$YOUR_APP_DNS/haproxy-status/)

# HAPROXY RAW DATA

[http://\\$YOUR\\_APP\\_DNS/haproxy-status;/csv](http://$YOUR_APP_DNS/haproxy-status;/csv)



# QUESTIONS?



# WANT TO LEARN MORE?

- Come hang out with us on IRC: [#openshift on Freenode](#)
- Link to these slides: <http://socuteurl.com/widdlepuppytail>
- Free hosting on OpenShift: [OpenShift Online](#)
- OpenShift source code: [OpenShift Origin](#)
- Red Hat Enterprise Support: [OpenShift Enterprise](#)

presented by:

[ryan jarvinen](#) / [@ryanj](#)

