Red Hat OpenShift

OpenShift POC Review

In this review, we take a look at Redhat's OpenShift Platform as a Service software. The goal of this is to see whether or not setting up a private cloud installation of this would be something that we could offer potential clients.

OpenShift looks to position itself as an alternative to other cloud-based PaaS systems like Google App Engine, Microsoft Azure, and Oracle's Cloud Platform. Being based on open source code that is available in github, Openshift can be used on Redhat hosted servers or installed on a private cloud setup in the user's own datacenter using either RHEL, Fedora, or as of late January, CentOS.

How it Works

At the infrastructure level, OpenShift runs two kinds of servers. The first is the application **node**, which is where all the end-user applications reside and run. The second server is the **broker**, which serves as the PaaS management system. It handles the creation and automation of all the end-user applications that are deployed to the node.

Within the application node, each user creates an application that resides in a container called a **gear**. Each gear is secured using SELinux and allocated resources via linux Control Groups. Openshift has 3 sizes of gears (small, medium, large.) When installing OpenShift on a private cloud, these can be given custom allocations. RedHat uses the following sizes in their OpenShift Online system for comparison.

- Small gears provide 512MB of RAM, 100MB of swap space, and 1GB of disk space
- Medium gears provide 1GB of RAM, 100MB of swap space, and 1GB of disk space
- Large gears provide 2GB of RAM, 100MB of swap space, and 1GB of disk space

The gears can be mixed and matched to run individual applications, or combined together to run different aspects of a single application, or be used to help scale the application when needed via HA.

Within each application, services like tomcat, php, ruby, JBoss EAP, mysql, postgresql, etc, are called **cartridges**. OpenShift comes with a good selection of built-in cartridges to get started. Custom quickstart applications, which are bundled applications that are ready to go like Wordpress and Drupal can also be installed from the web view links to the correct repos in github.

OpenShift also has what are called **Districts**. A district if a set of applications nodes within which gears can be reliably moved around to manage resources usage of the included nodes. Districts can be used to segregate gears of a defined size to a specific set of nodes. RedHat strongly advises that districts be enabled as it significantly reduces the number of potential problems that could arise from moving a gear to a different node.

Design Considerations

While all the Openshift components can be installed on a single physical server or virtual machine, Redhat recommends that the platform be spread out across at least 4 servers.

1. Broker
2. Applications Node (Or more depending on expansion)
3. MongoDB server
4. ActiveMQ server

As this is just a demo, I opted for slightly more condensed version of 1 broker node handling the management, MongoDB, and ActiveMQ servers with a separate VM to serve as the application node. In both configurations, this can lead to some rather large single points of failure as there is no redundancy between the management servers and the application servers, If the broker, or one of it's systems goes down, then OpenShift grinds to a halt. A more fault tolerant setup would need a little over twice as many servers at a minimum.

1. 2x Broker Virtual Machines with High Availability
2. 3x MongoDB / ActiveMQ Virtual Machines

3. 2x DNS Virtual Machines in a cluster (Assuming DNS is not being handled elsewhere already)
4. 2x Application Node (Physical Server)
5. 1x Load Balancer to handle HA capabilities for the two Brokers
   Reference: ose-installation-configuration-vfinal.pdf
6. 1x Authentication server (can be run from the Broker as well.)

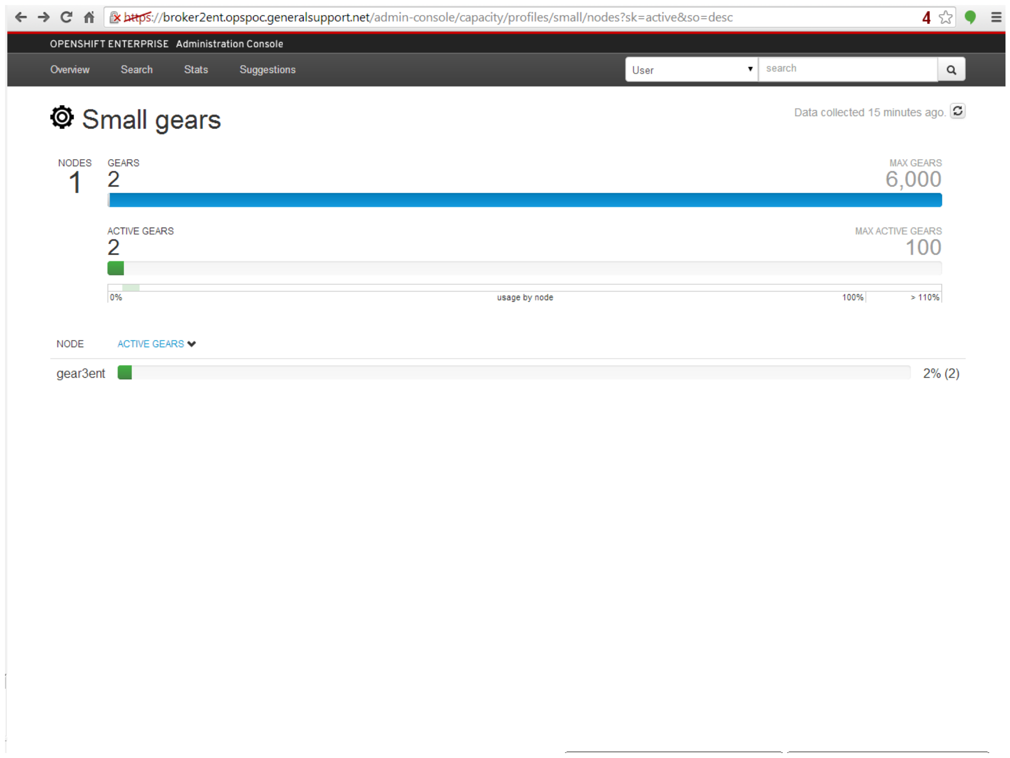
I have also set up two test platforms. One to test OpenShift Origin and one to test Openshift Enterprise.

Support

As OpenShift Origin is an opensource piece of software, there is a good deal of documentation located at http://openshift.github.io/documentation/ . IRC and the usual mailing lists are available as well. OpenShift Enterprise does not have much documentation as they want you to have a RedHat support plan since you are after all, using RHEL servers. Obviously, the cost to license the OS along with the price of a support plan may be a something to keep in mind.

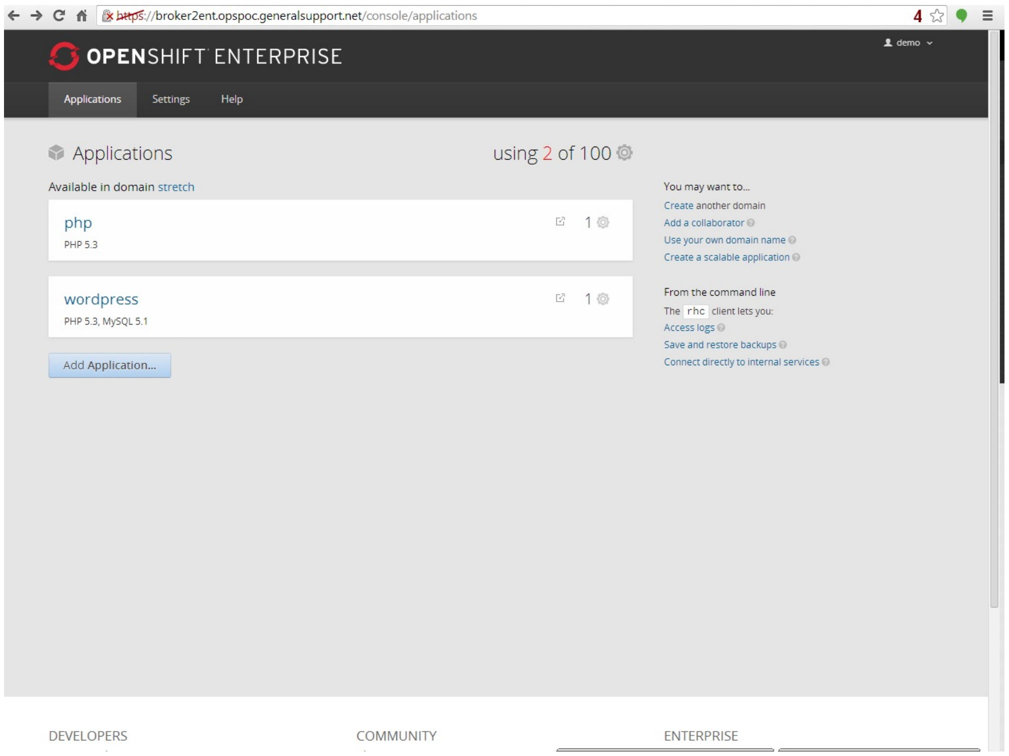
Administration

By default the admin console gui is limited in access to the broker host only. Adding a couple ProxyPass lines for the admin-console plugin and its assets in the apache config quickly gets around this however. Additionally, opening this up to be publicly accessible appears to imply that all users can access this, which can be a possible security issue (source.) At this time, the admin-console is really a read-only system that will aggregate data about the cloud, the users, and gears and give you charts documenting resource usage. It does have the ability to make suggestions on whether more resources are needed or if existing resources should be rearranged to better suit the current needs of the cloud. The bulk of the administration tools are found in shell and a lot of work for adding new users, and making config changes would be done on the broker node(s.) This is something I that I believe is a significant drawback as we would need to either manually add users and their limits all the time, or develop in house scripts to accomplish this.


~~https~~://broker2ent.opspoc.generalsupport.net/admin-console/capacity/profiles/small/nodes?sk=active&so=desc
OPENSHIFT ENTERPRISE Administration Console
Overview
Search
Stats
Suggestions
User
search
Small gears
Data collected 15 minutes ago.
NODES 1
GEARS 2
MAX GEARS 6,000
ACTIVE GEARS 2
MAX ACTIVE GEARS 100
0%
usage by node
100%
> 110%
NODE
ACTIVE GEARS
gear3ent
2% (2)

Usage

For the end-user, the OpenShift gui primarily exists to manage your installed applications and manage access to the account. Since each application in the gear has a built-in Git repository, most of the interaction with the the OpenShift system will be via git. The web gui can be bypassed completely by installing the OpenShift client tools on the local server that the user is developing on. This provides all the access of the gui, but in a terminal. If the user is used to git then this should be pretty easy to get the hang of.



Applications using 2 of 100

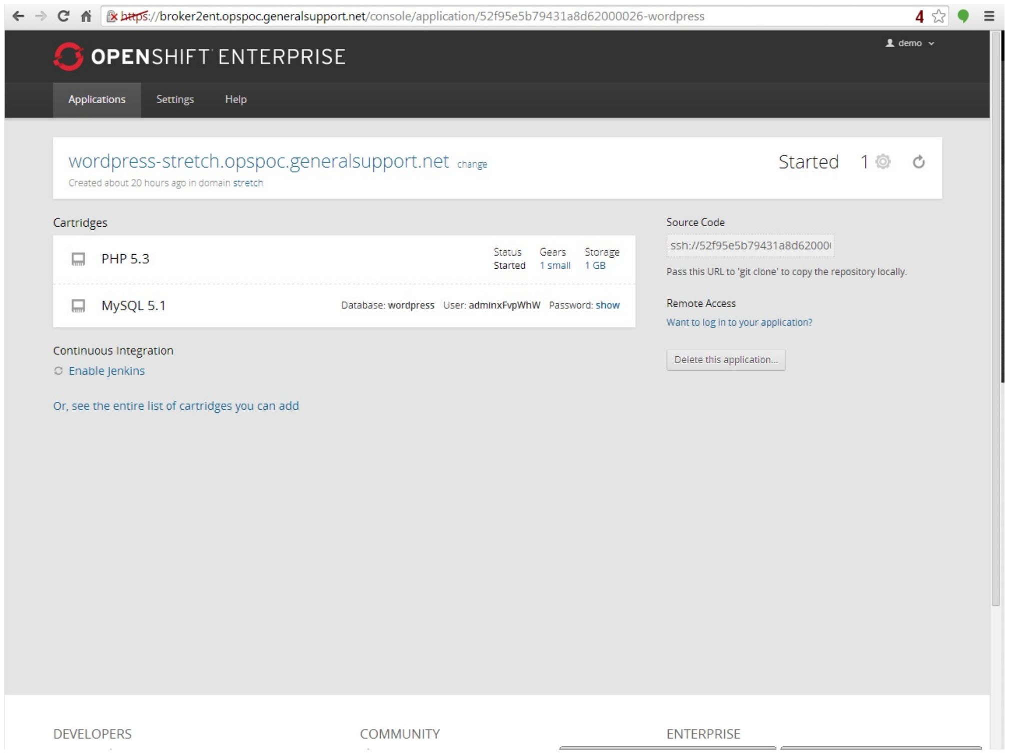
Available in domain stretch

php
PHP 5.3 1

wordpress
PHP 5.3, MySQL 5.1 1

You may want to...
Create another domain
Add a collaborator
Use your own domain name
Create a scalable application

From the command line
The `rhc` client lets you:
Access logs
Save and restore backups
Connect directly to internal services


https://broker2ent.opspoc.generalsupport.net/console/application/52f95e5b79431a8d62000026-wordpress
OPENSHIFT ENTERPRISE
demo
Applications
Settings
Help
change
Delete this application...
DEVELOPERS
COMMUNITY
ENTERPRISE

wordpress-stretch.opspoc.generalsupport.net
Created about 20 hours ago in domain stretch
Started 1

Cartridges

	Status	Gears	Storage
PHP 5.3	Started	1 small	1 GB

MySQL 5.1 Database: wordpress User: adminxFvpWhW Password: show

Continuous Integration
Enable Jenkins

Or, see the entire list of cartridges you can add

Source Code
ssh://52f95e5b79431a8d62000

Pass this URL to 'git clone' to copy the repository locally.

Remote Access
Want to log in to your application?

Benefits

- With it being open source and what looks to be a decent community backing behind it, I did find however that I had better luck looking for fixes for OpenShift Origin than I did for Enterprise, but that will be mentioned later on as a potential issue.
- Once it is up and running, the underlying cartridges that are powering the end-user applications will be updated automatically since they are installed as RPMs. The quickstart applications will not be updated automatically but that is not surprising since they are generally created by the community. When creating an application, OpenShift will tell you if the cartridge you are adding will be automatically

updated or not.
- While the interface is a bit spartan, this Is not a hard system for the end-user to pick up and run with. As mentioned previously, if the user is familiar with git already, then they should have no trouble jumping in and getting applications created and developed.

Potential Issues

- Applications are not set to be supported by HAproxy by default. That must be selected when the application is created. Additionally, HAproxy cannot be added to an application ex post facto. It appears that HAproxy will also take up its own gear so the end-user must be cognizant of these facts when creating an application and managing their gear usage.
- It seems like the turnaround time for adding new cartridges can be at times slow, and OpenShift Enterprise looks to really lag behind OpenShift Origin, though that is probably due to increased testing.
- In general, when trying out the various installation methods, the only one that provided no issues was the OpenShift Enterpirse template installation. This is most likely due to the fact that I was using RHEL for the operating system and RHN to download all the required packages. Aside from that test, every other test installation did not go smoothly. While most of the problems were of my own doing, subsequent installation tests seemed to show issues with the oo-install method. I was able to get around this using puppet and a config to deploy the broker and node. That being said, once a config is built, it could be used to deploy new nodes or brokers or other support servers as needed.
- I found the administration side of things to be lacking. The admin-console is a read-only gui and if it's allowed to be view externally, then everyone can see the stats. There are some APIs available to get that information, but RedHat warns that they are not set in stone and are subject to change,
- If you don't set up districts from the beginning, you run the risk that any gears that were created before the districts may not be able to be moved to other nodes.

Overall

While OpenShift looks like an interesting product to use, the numerous issues encountered when trying to set up the various test platforms, coupled with other complaints leads me to conclude that at this time, a production deployment would not be a worthwhile endeavor. Running a private version of OpenShift Enterprise would give us the most stability I believe, but it appeared to not have as many features out of the box of OpenShift Origin did and additionally would require OS and support licenses from RedHat. Conversely OpenShift Origin has more available applications that can be quickly installed, but it required a good amount of work to get up and running. It is possible that some of this is due to Official CentOS support for OpenShift being less than a month old at the time of writing, so there may be some kinks that they need to work out still. If the management support were to improve, then I could see this as being something to consider offering.

Quirks

When trying to deploy an instance of the wordpress or drupal quickstarts in OpenShift Origin, I discovered that I couldn't do it through the gui. The github repo for the wordpress quickstart, https://github.com/openshift/wordpress-example suggested to just create the application in shell

```
rhc app create wordpress php-5.3 mysql-5.1
--from-code=https://github.com/openshift/wordpress-example
```

Since that worked, I checked the quickstart json config file at */etc/openshift/quickstarts.json* and discovered that the quickstart (which is originally provided by RedHat,) was looking for the mariadb cartridge. This was the case for any included quickstart that required a database.

```
"quickstart": {
            "id": "8",
            "name": "WordPress",
            "website": "http://wordpress.org",
            "initial_git_url": "git://github.com/openshift/wordpress-example.git",
            "cartridges": ["php-5", "mariadb"],
            "summary": "A semantic personal publishing platform written in PHP with a
MySQL back end, focusing on aesthetics, web standards, and usability. Administrator
user name and password are written to $OPENSHIFT_DATA_DIR/CREDENTIALS.",
            "tags": ["php", "wordpress", "blog", "framework", "instant_app"],
            "admin_tags": []
        }
```

Obviously documentation is still lacking in some parts, though this was quickly remedied by changing the json file to use mysql and clearing the broker cache.