OpenShift and Cloud Foundry PaaS:
High-level Overview of Features and Architectures

by Alexander Lomov, R&D Engineer at Altoros
Table of Contents:

1. Executive Summary .................................................................................................................. 3
2. The History of OpenShift and Cloud Foundry ....................................................................... 3
3. Overview of Supported Features ............................................................................................. 4
4. PaaS Installation ...................................................................................................................... 5
   4.1 OpenShift .......................................................................................................................... 5
   4.2 Cloud Foundry .................................................................................................................. 5
5. Architecture Comparison ......................................................................................................... 6
   5.1 Similar components .......................................................................................................... 6
   5.2 The architecture of OpenShift ......................................................................................... 6
   5.3 The architecture of Cloud Foundry .................................................................................. 8
6. Conclusions ............................................................................................................................. 8
7. About the Author .................................................................................................................... 10
1. Executive Summary

Among all open source projects in the category known as Platform-as-a-Service, OpenShift and Cloud Foundry have amassed the strongest development communities. Many consider them to be the leading open source PaaS. With similar functionality and goals, both make it possible to write code in a variety of languages and deploy applications to public or private clouds. Both are evolving extremely fast. Still, few, if any, in-depth comparisons of OpenShift and Cloud Foundry exist.

The purpose of this research is to provide a high-level overview of Cloud Foundry and OpenShift, side by side. Any feedback from the members of the community in regards to the accuracy of data contained in this white paper is welcome. Your opinion will help to make this and future comparisons as fair as possible.

This is only the first version of the document; depending on what will happen to these two platforms in the months coming after the Cloud Foundry Foundation announcement, an extended paper may be published later. It will compare adoption, features, operational capabilities, continuous delivery, application life-cycle management, architecture, deployment principles, programming models, and developer communities for both platforms. The ultimate goal is to provide enterprise architects, IT managers, and members of the vendor ecosystems with a PaaS comparison framework—based on a competitive scorecard—that will help to compare these PaaS products for private deployments within an enterprise.

(Disclosure: Altoros is a member of the Cloud Foundry developer community.)

2. The History of OpenShift and Cloud Foundry

Cloud Foundry

The story of Cloud Foundry starts in summer of 2007 when Chris Richardson, a software developer and entrepreneur, created the Cloud Tools project. Initially, this was a set of tools for deploying Java applications to Amazon EC2. In 2008, Chris launched Cloud Foundry, a commercial version of Cloud Tools. Inspired by success of Richardson’s initiatives, SpringSource (led by VMware by that time) acquired Cloud Foundry in 2009.

New era of the product began on April 12, 2011 with an open source release of the Cloud Foundry PaaS from VMware. In 2013, VMware and EMC formed Pivotal, a provider of application and data infrastructure software, Agile development services, and data science consulting. Currently, Pivotal is the main contributor to Cloud Foundry, which is also supported by large market players (such as IBM, Cisco, and others). This PaaS managed to gather a very enthusiastic community that is actively contributing to its development.

Today, commercial PaaS systems based on Cloud Foundry are available from several vendors: ActiveState (check out their Stackato), Uhuru, CenturyLink (which recently acquired Tier3 and AppFog, providing now solutions based on Cloud Foundry), and others. Additionally, the Ubuntu community has announced a Cloud Foundry-based solution on top of OpenStack as Ubuntu’s direction for PaaS. Finally, on November 12, 2013, Pivotal introduced their new enterprise PaaS built on top of Cloud Foundry.

OpenShift

One month after the release of Cloud Foundry, on May 4, 2011, Red Hat announced their own open source cloud platform, OpenShift. The PaaS is developed and supported by Red Hat and its community. From the beginning, OpenShift was treated as the main core for Red Hat’s Platform-as-a-Service. Currently, several options exist: open source OpenShift Origin (distributed under the Apache License); OpenShift Online, a commercial PaaS for hosting applications in a public cloud; and the Enterprise version for private clouds.
3. Overview of Supported Features

As of January 2014, both systems have a lot in common:

- both can serve as a private PaaS
- both are built on open source technologies and distributed under the Apache License 2.0
- both support a broad range of popular development languages and frameworks
- both platforms can be hosted on most popular IaaS, such as AWS and OpenStack

The differences between OpenShift and Cloud Foundry can be summed up in the following table:

Table 1: Comparing the features available in OpenShift and Cloud Foundry. Source: Altoros

<table>
<thead>
<tr>
<th>Feature</th>
<th>OpenShift (open source version)</th>
<th>Cloud Foundry (open source version)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web site</td>
<td>openshift.com</td>
<td>cloudfoundry.org</td>
</tr>
<tr>
<td>Source code</td>
<td>openshift.github.io</td>
<td>github.com/cloudfoundry</td>
</tr>
<tr>
<td>First release</td>
<td>May 4, 2011</td>
<td>April 12, 2011</td>
</tr>
<tr>
<td>Possible workflows and available tools</td>
<td>Managing cloud and application deployment through an IDE, a Web console, or a command line. These are built-in instruments provided out-of-the-box.</td>
<td>IDE integration support, at least for Eclipse and IDEA, as well as command line tools and Ruby support through the Ruby Gem called cfoundry. Web Console projects are developed by volunteers of the open source community (for instance, CF-Console).</td>
</tr>
<tr>
<td>Written in</td>
<td>Ruby, C, and Node.js</td>
<td>Ruby, C, Go, and Java (the UAA component)</td>
</tr>
<tr>
<td>Main developer (besides the community)</td>
<td>Red Hat, Inc.</td>
<td>Pivotal Software, Inc.</td>
</tr>
<tr>
<td>Top community supporters</td>
<td>Red Hat, SPARC, AppDynamics, BitRock, and Kinvey</td>
<td>Pivotal, IBM, CenturyLink, General Electric, ActiveState, Intel, Piston Cloud, Altoros, HP, SAP, Rackspace, and others</td>
</tr>
<tr>
<td>Deployment types</td>
<td>1) Code push 2) Deployment through repository (allows for having a repository on your PaaS, just like Heroku)</td>
<td>Code push (deployment through code copying)</td>
</tr>
<tr>
<td>Supported languages</td>
<td>Java, Ruby, Python, PHP, Perl, JavaScript, etc.</td>
<td>Java, Python, PHP, Clojure, Scala, Erlang, Ruby, JavaScript, and all the languages provided by Heroku buildpacks</td>
</tr>
<tr>
<td>Supported services</td>
<td>MySQL, PostgreSQL, MongoDB, and SQLite (read this article)</td>
<td>MySQL, PostgreSQL (vFabric Postgres is supported, as well), MongoDB, and Redis</td>
</tr>
<tr>
<td>You can also extend OpenShift to support a technology you need by developing cartridges (for more, see this article).</td>
<td>You can also add support for custom services (common notes can be found here, see notes for Cloud Foundry v.2 here). The Cloud Foundry community provides most extensions, which can be downloaded from the GitHub repository.</td>
<td></td>
</tr>
<tr>
<td>Backup</td>
<td>Manual</td>
<td>Manual</td>
</tr>
<tr>
<td>Major supported frameworks and tools</td>
<td>Spring, Git, Zend Server for PHP, Codelgniter, Ruby on Rails, Django, Rack, Node.js, Flask, etc.</td>
<td>Ruby on Rails, Sinatra, Spring, Node.js, Play 2.0, Lift, Scala, .NET (via ironFoundry), etc.</td>
</tr>
<tr>
<td>Supported IaaS (default)</td>
<td>AWS and OpenStack</td>
<td>AWS, OpenStack, and vSphere</td>
</tr>
</tbody>
</table>
## Auto scaling

The feature is available out-of-the-box. This article describes how simple it is to set up auto scaling. It is a relatively new functionality that was added in summer 2013.

Not enabled out-of-the-box, but can be implemented with the RightScale platform (see this article).

## Configuration and management efforts

There are several options:

1) If you install OpenShift Origin on a private cloud, it will require engineers with specific expertise in the Red Hat infrastructure.

2) If you purchase OpenShift Enterprise or OpenShift Online as a service (a commercial product), Red Hat will handle all the management and configuration for you.

You will need a dedicated specialist to support a private Cloud Foundry deployment.

Unlike OpenShift, Cloud Foundry does not require the IT team to use a particular operating system.

The commercial version of the PaaS (managed by Pivotal) is currently in beta and can be accessed here.

## Hot deploys (zero-downtime deployment)

Available out-of-the-box

Although hot deploys are not available out-of-the-box, they can be implemented using third-party open source tools. For more, see this article.

## cron and background processing

Available

Available (more details here and here)

## WebSockets

Available

Available

## Pricing (as of Dec 2013)

**OpenShift Origin** is a downloadable version that can be used for private PaaS deployments.

**OpenShift Online** is a paid service; however, in addition to paid plans, it provides a free account that includes three small gears, each with 512 MB of RAM and 1 GB of storage. The Silver Plan costs $20 per month, providing 3 small gears with a possibility to add more for $0.02 to $0.10 per hour.

The pricing for **OpenShift Enterprise** is available upon request.

The open source version of Cloud Foundry is distributed free of charge.

The **Pivotal CF** hosted solution costs $0.03/hour per 1 GB of an application instance (this includes AWS hosting charges). You can use 25 application services for free or add more in the Premium plan.

Proprietary Cloud Foundry distributions are also available from AppFog, Uhuru, ActiveState, and others.

## 4. PaaS Installation

### 4.1 OpenShift

Though we succeeded in installing OpenShift, there were some issues when setting it up using the official documentation and installer. It turned out that the Vagrant file had an outdated URL leading to a virtual machine box. Additionally, the standard image had not been configured to work with the installer. Anyway, we succeeded in installing OpenShift using Vagrant and Puppet as described here (having submitted the bug to the OpenShift tracker).

### 4.2 Cloud Foundry

Using Vagrant and VirtualBox, we installed Cloud Foundry with the BOSH-lite tool chain on a local host without any issues. This approach will work for those who want to get a full-blown PaaS with services and...
have enough RAM. Other methods also exist. For instance, if you need a sample Cloud Foundry instance (i.e. you do not want the services right away), you can consider the CF Nise Installer from NTT Labs.

5. Architecture Comparison

5.1 Similar components

Both OpenShift and Cloud Foundry are focused on delivering Web-based applications and feature components with similar functionality:

- **Routers** manage user traffic.
- **Working Nodes** are used to run Web applications.
- **Managers** manage and monitor **Working Nodes** and take care of them in case of failures.
- **The Messaging Bus** enables collaboration between different parts of a distributed PaaS.

*Table 2. The major components of OpenShift and Cloud Foundry. Source: Altoros*

<table>
<thead>
<tr>
<th>Component / function</th>
<th>OpenShift</th>
<th>Cloud Foundry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router</td>
<td>Brokers, HAPProxy Gears</td>
<td>Router</td>
</tr>
<tr>
<td>Working nodes</td>
<td>Gears</td>
<td>Warden containers within DEA</td>
</tr>
<tr>
<td>Messaging bus</td>
<td>ActiveMQ</td>
<td>NATS</td>
</tr>
<tr>
<td>Managers</td>
<td>Broker Support Nodes</td>
<td>Cloud Controller</td>
</tr>
<tr>
<td>Providing resources and services to applications</td>
<td>Cartridge</td>
<td>Buildpacks and services</td>
</tr>
</tbody>
</table>

Let's review both systems in detail.

5.2 The architecture of OpenShift

OpenShift uses unified abstractions to work with applications in the cloud. **Gears, Brokers, and Cartridges** are the core concepts of this PaaS.

**Gears** are implemented as lightweight containers with separate namespaces and SELinux modules. Designed to run applications, they have separated access to shared resources. Using Linux control groups, a fair share of the CPU, RAM, disk space, and network bandwidth is allocated to each gear.

**Cartridges** provide the actual functionality necessary to run a user application. They add support for programming languages and access to various databases. In fact, cartridges are add-ons that contain binaries, setup, and control scripts that make it possible to deploy and maintain the functionality of applications. They are also responsible for setting up environment variables to bind gears with the necessary resources.

**Brokers** are the point of contact for all application management activities and traffic. They are implemented as daemons responsible for managing user logins, DNS, and application status. The OpenShift cloud platform always works through a broker using a REST-based API.

**Nodes and Broker Support Nodes (BSN)** represent the lower layer of the OpenShift architecture. Nodes are the physical machines where the gears are allocated. BSN are the nodes that run brokers. BSN and nodes are connected through a **Messaging Bus**. For this purpose, OpenShift uses ActiveMQ, a tool that has existed for a while, having passed the test of time. It also has a Web console, but you must have the necessary experience to understand the information it provides.

+1 (650) 395-7002
e-mail: engineering@altoros.com
www.altoros.com | twitter.com/altoros

© 2014 Altoros Systems, Inc. Any unauthorized republishing, rewriting or use of this material is prohibited. No part of this resource may be reproduced or transmitted in any form or by any means without written permission from the author.
OpenShift implements horizontal scaling using gears with HAProxy. Brokers make decisions based on messages from the HAProxy and gears do the scaling. Official documentation does not provide any details, but the functionality is available, nevertheless.

Figure 1. OpenShift components and their interactions

The configuration that you can see on Figure 1 shows how OpenShift components interact. Developers can work with OpenShift through a broker application with a REST API client or using the REST API directly. The broker interacts with the nodes through a messaging bus and is also responsible for authentication, DNS management, and data storage services. The nodes contain gears with application code repositories. The gears are configured by cartridges that are connected to them. All traffic generated by users of the apps deployed in the cloud passes through a proxy server that distributes calls between the gears. In the use case shown on Figure 1, the user is forwarded to an HA Proxy (Gear 1) that servers as a load balancer for the application deployed on gears 2 and 3. Application gears work with the database through the environment provided by cartridges (for instance, a PostgreSQL cartridge binds the application with PostgreSQL server inside OpenShift cloud).

In 2013, Red Hat did a lot to add support for Docker—a container orchestration layer built on top of LXC (which, in its turn, is implemented using cgroups, Linux control groups). For the same purpose, Cloud Foundry uses Warden, which provides a different way to manage containers (see this comparison of Warden and Docker). However, ActiveState already included Docker in their Stackato PaaS based on Cloud Foundry v2. There are chances that the Cloud Foundry project might do the same in the future.
5.3 The architecture of Cloud Foundry

Cloud Foundry has more direct and straightforward abstractions than OpenShift (see Figure 2). The dynamic routing layer (Router) handles all the traffic. It resolves application traffic and developers’ Cloud Foundry REST API requests. The Cloud Controller component is responsible for all management tasks. Being the main end point for the Cloud Foundry REST API, it uses the UAA module to authenticate and authorize users. Health Manager monitors the status of applications and takes appropriate actions, when it changes. For instance, if an application instance is down, it automatically triggers the system to create a new one. The NATS messaging system processes notifications. Services provide everything that an application may need. It is possible to bind (associate) an application with any existing or new services—e.g., the PostgreSQL DB, the RabbitMQ message broker, etc.

Figure 2. The architecture of Cloud Foundry. Source: Cloud Foundry documentation

Cloud Foundry uses Warden to manage containers. Warden allows for manipulating application containers based on Linux control groups (cgroups).

BOSH is the recommended tool for deploying and running Cloud Foundry. It is a tool chain for orchestrating and managing deployments of complex distributed systems, including various databases. BOSH can create cloud setups on popular IaaS and support their operation using a deployment manifest (a file that describes the necessary services and their configuration). It also has the tools to monitor cloud status, fix errors (for instance, recreate virtual machines that have failed), and update versions of the packages included into the deployment.

Those who would like to try Cloud Foundry on a local machine, can use BOSH-lite, a local development environment for BOSH that uses Warden containers in a Vagrant box.

One of the greatest advantages of Cloud Foundry is that it supports buildpacks designed for Heroku. Buildpacks can deploy applications, as well as install all the application dependencies. When you push an application, the system automatically applies the appropriate buildpack. For instance, this buildpack adds support for Ruby applications.

6. Conclusions

The last couple of years have demonstrated the benefits of cloud automation. The emergence of OpenShift and Cloud Foundry became the next step in this field. Several years ago, only public PaaS were available (such as Google App Engine and Heroku). Now there are a number of open source solutions for creating a
private enterprise PaaS. The stakes are high: according to IDC, the PaaS market will reach $14 billion in 2017.

With the same goals, both OpenShift and Cloud Foundry are evolving very rapidly, supported by the open source community and large vendors. For instance, there were over 15,000 commits from 732 contributors to Cloud Foundry in 2013 (as indicated on the CF community Web site). No precise data has been disclosed for OpenShift, but the development process is also very active. Nevertheless, there are some essential differences between the two systems:

- OpenShift works on the Red Hat operating system only, while Cloud Foundry can run on all major Linux-like systems and has interfaces for most popular IaaS providers.
- Cloud Foundry natively supports Heroku buildpacks, a large collection of ready-to-use services. Though OpenShift uses cartridges for the same purpose, migration from a public Heroku platform to a private PaaS will not be as easy.
- Cloud Foundry has good integration with VMware’s products, including support for vCloud migration and the vSphere hypervisor. In its turn, OpenShift is well-integrated with Red Hat’s products.
- Cloud Foundry uses Warden containers, while OpenShift uses Docker containers, which have a different kind of abstraction.
- Out-of-the-box, OpenShift supports deploying applications through a Git repository, hot deploys, and auto scaling. Cloud Foundry does not support these features out-of-the-box, but they can be enabled using open source third-party tools.
- OpenShift is mainly developed by Red Hat, while a number of tech leaders are working on Cloud Foundry, aiming at making it a de-facto industry standard.

Obviously, those using Red Hat Linux will probably still go for OpenShift, while Ubuntu users are likely to choose Cloud Foundry. Canonical has already announced that the new release of Ubuntu will include a Cloud Foundry-based PaaS for OpenStack. This will probably result in a faster PaaS, better optimized to work with Ubuntu.

On the other hand, Gartner has recognized OpenShift in its Magic Quadrant for Enterprise Application PaaS (Jan 2014). It is hard to say if these results are representative, since they are based on data before July 31, 2013. Since then, Pivotal has launched Pivotal One, an enterprise-ready PaaS, which is very likely to get into Gartner’s 2015 Magic Quadrant. On top of that, each month it announces several new large-scale enterprise customers or partners adopting Cloud Foundry as a PaaS. The launch of the Cloud Foundry Foundation on Feb 24, 2014, changes the game for the entire PaaS industry.

Depending on what will happen to OpenShift and Cloud Foundry in the coming months, we might have to update this research later. In this case, we will provide a deeper investigation of the platforms. Feel free to contact us if you would like to get these results before they are published online.
7. About the Author

**Alexander Lomov** is an R&D Engineer at [Altoros](www.altoros.com), a company that builds Platform-as-a-Service solutions. With extensive Ruby experience, Alexander is a fan of open source, having contributed to Fog, RefineryCMS, Netzke, simple_form, and other projects. His professional interests include AWS, MySQL, PostgreSQL, Cassandra, MongoDB, C++, jQuery, JavaScript, and much more.

[Altoros](www.altoros.com) provides consulting and managed services around Cloud Foundry PaaS, multi-cloud deployment automation, and complex Java/.NET/Ruby architectures that make heavy use of SQL/NoSQL/Hadoop clusters. Altoros employs 250+ senior and mid-level engineers across 7 countries, including one of the largest pools of Cloud Foundry expertise on the market. Fast-track services (4–12 weeks) include single-cloud deployments of Cloud Foundry and integration with relational and NoSQL databases. For more, please visit [www.altoros.com](www.altoros.com) or follow [@altoros](https://twitter.com/altoros).