

VMware's Cloud Foundry Flounders

July 2011

VMware (www.vmware.com) is the new kid on the block when it comes to cloud computing. Its offering, the Cloud Foundry, is aptly named since it is not simply another cloud service. Rather, it is intended to provide a platform for developers to build their own clouds. VMware's intent is that the Cloud Foundry will become the "Linux Kernel of Cloud Operating Systems."

VMware launched the Cloud Foundry on April 12, 2011. It did so by offering free access to its own beta version of a PaaS (Platform as a Service) cloud for developers to experiment with their own clouds.

Unfortunately, just two weeks after its launch, the Cloud Foundry suffered a sequence of major problems that took it down for about twelve hours over a day and a half period. Coincidentally, this was just five days after Amazon suffered a devastating crash that took down its EC2 (Elastic Compute Cloud) cloud for four days and resulted in some irretrievable data loss.¹ Interestingly, both outages entailed a storage subsystem fault that was subsequently aggravated by a human fat finger.

Before we get into what happened to VMware's Cloud Foundry, let's look at what the Cloud Foundry is.

The Cloud Foundry

The Cloud Foundry is free, open-source cloud-computing software developed by VMware. It can support multiple frameworks, multiple cloud providers, and multiple application services, all on a cloud-scale platform.

Cloud Foundry Offerings

The Cloud Foundry is offered in three different forms:

- CloudFoundry.com is a complete, multitenant PaaS (Platform as a Service) environment hosted and managed by VMware. It runs on VMware's vSphere cloud platform and is written in Ruby. Currently, CloudFoundry.com supports Spring for Java apps, Rails and Sinatra for Ruby apps, Node.js apps, and apps for other JVM frameworks, including Grails.

¹ Amazon's Cloud Downed by Fat Finger, *Availability Digest*, May 2011.
http://www.availabilitydigest.com/public_articles/0605/amazon_ebs.pdf

CloudFoundry.com is in beta and can be accessed for free. Users of the beta service will be notified regarding the end of the beta phase. The pricing for commercial service has not yet been announced.

- CloudFoundry.org is an open-source project where developers and community members collaborate and contribute to the project. It is released as free software under the terms of the Apache License 2.0.
- **Cloud Foundry Micro Cloud** is a single developer instance of Cloud Foundry. It provides developers with a personal PaaS that runs on their desktops. Micro Cloud is provided as a downloadable software image for VMware Fusion or VMware Player as well as a hosted image on selected cloud partners. Yet to be released, Micro Cloud will be a free download for developers interested in a single instance, personal PaaS on their local machines.

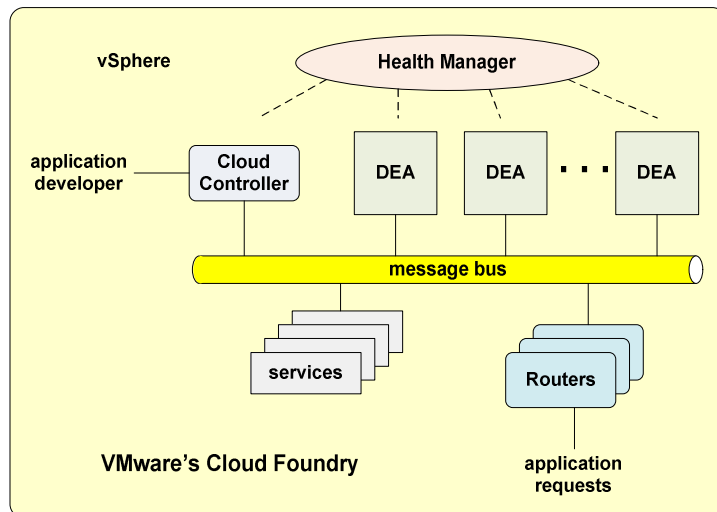
Cloud Foundry Architecture

The Cloud Foundry platform consists of six major components – Droplet Execution Agents (DEAs), a Cloud Controller, a Health Manager, Routers, services, and a message bus. Virtualization services are provided by the underlying VMware vSphere cloud platform.

Droplet Execution Agent

The Droplet Execution Agent, as its name implies, executes *droplets*. A droplet contains all of the objects that comprise an application. These objects are organized into a tar² file and are wrapped with a pair of start/stop scripts. The droplet also has one input that is used by the DEA to assign a port number over which the application can receive requests.

There can be several server nodes in a Cloud Foundry environment. A DEA runs on each node and is responsible for running the applications assigned to its node.



When a new application is ready to be deployed, the DEAs receive a broadcast message from the Cloud Controller (CC - described next) announcing the availability of the application. If a DEA has the resources to execute the application, it responds affirmatively. If it is chosen by the CC, it will receive the application's droplet from the CC.

Once it receives a droplet, the DEA calls the droplet's start script to start the application. The DEA uses the droplet's input to inform the application on which port number to listen for requests. The application then broadcasts its location over the message bus so that the Routers know of its existence.

² [tar \(file format\)](#), *Wikipedia*.

Cloud Controller

The Cloud Controller (CC) is the developer's interface to the Cloud Foundry. When a developer is ready to deploy an application, he submits a directory tree of his application files to the CC. The CC checks to see what files it already has and returns a list of the files that it needs in order to be able to build the application.

Once it has all of the components it needs, the CC creates a droplet for the application. Once it has packaged the application into a droplet, it broadcasts on the message bus that it has an application to run. Based on the responses received from the DEAs, it will assign the application to one or more DEAs for execution by sending them the droplet.

Routers

A Cloud Foundry platform has a pool of identical Routers. The Routers receive application requests from external entities and route them to the addressed applications.

The Routers listen to DEA broadcasts to know which applications are coming online and which are going offline. When an application request is received by the Cloud Foundry, it is routed to one of the Routers via a load balancer. The Router determines the application to which the request is to be routed based on the request's header and sends the request to that application. If there are multiple instances of the application, the Router will choose one and will send the request to it.

Health Manager

The Health Manager is a stand-alone daemon responsible for the reliable operation of the Cloud Foundry. It periodically scans the CC's database to see what the state of the cloud should be. It then inspects the real state to see if that state matches the expected state.

If there is a discrepancy, the Health Manager will notify the CC. If the discrepancy is a downed application, the CC will spawn a new instance of the application. If a DEA node is down, the CC will redistribute the applications that had been running on that node to surviving DEAs.

Services

A set of services is provided for the applications to use. Included are messaging, caching, and database (MySQL) services, among others. It is VMware's intention that the set of services ultimately becomes a huge ecosystem provided by VMware and the open community.

Message Bus

The message bus provides a decoupled mechanism for all of the Cloud Foundry components to communicate with each other.

The Outage

The outage that occurred on April 25th and lasted until April 26th had to do with VMware's Platform as a Service offering (CloudFoundry.com) and consequently affected developers trying to build their own clouds.

According to a detailed account³ posted on April 29th by Dekel Tankel, one of the primary builders and managers of Cloud Foundry, the following sequence of events occurred:

- 5:45 am PDT, April 25th

The Health Manager detected intermittent failures in two DEA nodes. It reported these failures to the Cloud Controller, which took appropriate action to correct the problems. Over the next thirty minutes, the failures intensified and spread to other DEA nodes. The volume of failures started to impact the performance of the Cloud Controller.

- 6:11 am PDT, April 25th

The Health Manager suddenly issued a large number of alerts. All eight Cloud Controllers had lost their connections to a portion of the storage subsystem. This caused the controllers and the Health Manager to enter read-only mode.

As a result, all developer control operations became inoperative. These functions included logon, logoff, create application, start application, stop application, and others. However, existing applications continued to run normally.

- 3:30 pm PDT, April 25th

The problems were corrected, and the developers could once again log in and continue their development. They had been locked out of the Cloud Foundry for more than nine hours.

- The root cause of the failure

It was subsequently determined that the problems started with a partial outage of a power supply in one of the storage cabinets. This impacted a single LUN (logical unit number – a disk or set of disks) and made it inaccessible.

Such a failure was expected to happen, and the platform had been designed to recover from it by using redundant hardware and software. However, the monitoring system, the recovery software, and the operational procedures were not consistent with each other. Consequently, the fault was not properly handled, causing the Cloud Controller to declare a loss of connectivity with a section of storage.

It took several hours to verify that there was no loss of data and that the storage subsystem was operating properly before service to the developers was restored.

- 8 am PDT, April 26th

To prevent this situation from happening again in the future, it was decided to develop a full operational playbook to properly guide Cloud Foundry operating staff in the early detection, prevention, and restoration of services should the platform fail to properly handle any loss of connectivity to storage.

Explicit instructions were issued not to make any keyboard entries until the playbook had been formally reviewed by the engineering and operations team in a meeting scheduled for noon. This was to be a hands-off the keyboards, paper-only exercise until the playbook review was complete.

³ [Analysis of April 25 and 26, 2011 Downtime](#), *Cloud Foundry Community*; April 29, 2011.

- 10:15 am PDT, April 26th

Unfortunately, one of the operations engineers “touched” a keyboard.⁴ This resulted in a full outage of the network structure that front-ends the Cloud Foundry. It took out all the load balancers, the routers, the firewalls, and other network components as well as a portion of the DNS infrastructure. There was a complete loss of external connectivity to the cloud platform.

All of the applications in the DEA nodes continued to run, but it was only the operations staff that knew this. No one could access them from the outside.

- 11:30 am PDT, April 26th

The operations staff successfully restored the network. The Cloud Foundry was once again up and running.

Dekel Tankel ended his analysis by saying:

“We take full responsibility for these issues and apologize to our users who were impacted by them. We can and will do better, having already learned from these incidents. We greatly appreciate your patience as we improve our service and the underlying technology, while building capacity to deal with the extraordinary level of demand that we are experiencing.”

- The Cloud Foundry Team

Lessons Learned

VMware’s outage did not have the same negative effects on its users as did the Amazon outage a week earlier. The Cloud Foundry is a beta release that is expected to have problems – that is why we have beta releases. Users have free access to it and are penalized only by their inconvenience. This outage was simply a case of infant mortality. Nevertheless, it is important to learn from this experience.

Despite the rather full disclosure made by VMware following this series of outages, there are several curious questions still remaining.

- How could a power-supply failure affect a portion of a storage subsystem? Were there not redundant power supplies?
- Why did the lack of access to a LUN in the storage unit take down the Cloud Controller? Where was the backup database copy?
- What possible sort of keyboard action could have wiped out the entire network feeding the cloud?

Clearly, VMware has a lot to learn about managing a cloud before it goes prime time. However, experiences like this are the best teacher. One of the key lessons that VMware must have learned from this incident is that proper documentation, testing, and training is absolutely necessary to ensure that operational procedures for any purpose are adequate.

⁴ “touched” is the term used in the analysis blog. There were large outcries on many blogs as to what “touched” meant. A simple brush of a keyboard with a hand? An inadvertent depression of the Enter key, thus executing a command that had earlier been entered but not executed? The entry of an actual command?

One thing that VMware did that was admirable was its transparent description of the events that led up to the outages and their root causes. This is often a failing of an organization, leading to a great deal of frustration for its users. In fact, VMware seemed to beat the press. The press accounts describing these incidents didn't start until about the first of May, a week later.

One subtle lesson that cloud outages such as those experienced by Amazon and VMware teach is that the more resources you try to bring together, the more things there are that can go wrong. This is bound to be a continuing lesson in cloud computing.

Acknowledgements

In addition to the references noted above, information for this article was taken from the following sources:

VMware Cloud Foundry Suffers Service Outage, *Information Week*, May 4, 2011.

Cloud Foundry Open PaaS Deep Dive, *cloudfoundry.com*.

Cloud Foundry, *Wikipedia*.

Cloudfoundry.com