The Little Cloud Foundry Book

Jonathan "Duke" Leto

Leto Labs LLC

About This Book

License

About The Author

Jonathan ``Duke'' Leto, Founder Leto Labs LLC

Thanks, Y'all

…

Latest Version

The latest source of this book is available at:

http://github.com/letolabs/the-little-cloud-foundry-book.

Who Is This Book For?

This book is for three sets of people:

- Application developers wanting to run code *on* Cloud Foundry
- Dev-Ops people who want to install, maintain and manage Cloud Foundry instances
- Developers wanting to contribute, fix bugs and hack on Cloud Foundry itself

If you are an application developer, start from the beginning of this book. If you are a dev-ops person who is already familiar with running applications on CF, you can skip to the ``Installation'' chapter.

Developers who are familiar with Cloud Foundry but want to know how to contribute can skip to ``Hacking on Cloud Foundry''.

Introduction

Graecum est; non legitur -- ``It's Greek to me''

What is a Platform-As-A-Service (PaaS)?

A PaaS is middleware between the low-level infrastructure (i.e. the actual CPUs, hard-drives, RAM, etc) and the high-level applications that run on top of the platform. The boundary between the platform and the application is clear, but the boundary between the infrastructure and the platform is still largely in flux.

```
-------------------------
Software-as-a-service
-------------------------
Platform-as-a-service
-------------------------
Infrastructure-as-a-service
-------------------------
```

While there are many different murky definitions of PaaS floating around the market currently, there are a few functions that are usually the task of PaaS:

- Deploying new code to your application
- Allocating application services + resources dynamically
- Application health monitoring
- Load balancing

What is Cloud Foundry?

The name ``Cloud Foundry'' can be used to refer to either the commercial VMware product at cloudfoundry.com or the Cloud Foundry Open Source Project. In this book, ``Cloud Foundry'' will always refer to the open source project, unless otherwise noted.

Cloud Foundry consists of a large amount of Ruby code along with a small amount of C, and shell scripts that run on top of Ubuntu LTS. The C code (Warden) is used to manage Linux containers at the kernel level using a similar strategy as the Linux kernel user space program ``lxc''. CF also consists of many configuration files in YAML format.

The vmc client is written in Ruby and speaks JSON to a Cloud Foundry-compatible HTTP end-point. It is at this layer which many companies are competing for your business. Thankfully, almost all have free for non-commercial use, academic, non-profit and other free tiers, so you can ``test drive'' their flavor of the cloud before forking over your money.

A common misconception is that VMWare Cloud Foundry needs to be run on VMware infrastructure. That is not the case. While VMware vCloud is supported, Amazon (and compatible

APIs) are also supported out of the box.
Features

Supported Languages
Supported languages in mainline:

- Ruby (MRE 1.8.x, MRE 1.9.x, JRuby)
- Python
- PHP
- NodeJS 0.4.x, 0.6.x
- PHP 5.3, 5.4 (ActiveState, AppFog, paas.io)
- Erlang
- Java and JVM-based languages, such as Groovy

Supported languages in forks:

- Perl 5 (ActiveState)
- .NET (IronFoundry)
- .NET (Uhuru)
- Haskell 2011.4.0 (paas.io)

Supported Frameworks
Mainline:

- Rails
- Sinatra
- Rack (contributed recently by paas.io)
- Java Spring

Forks:

- Catalyst (ActiveState)
- Dancer (ActiveState)

Supported Services
Mainline:

- PostgreSQL
- MySQL
- redis
- neo4j
- RabbitMQ
- MongoDB
- vblob (S3-compatible HTTP endpoint)

Forks:

Currently awaiting data.

As for missing services in mainline, I would greatly like to see CouchDB supported.

Deploying your App to a Cloud Foundry Instance

If you just want to deploy apps to a Cloud Foundry instance, you just need to install a single Ruby gem called ``vmc''

```
gem install vmc
```

At this point you need to decide which Cloud Foundry based service to use. There are many. For now, if you want to register a free account at cloudfoundry.com, you can do that from the command line:

```
vmc register
vmc login
vmc info
```

How Do I Choose a Cloud Foundry Provider?

This question depends on these questions in the following order:

- Do you need complete privacy?
- Do you want to DIY it or pay somebody else to worry?
- Which languages do I need supported?
- Do I need value-add services such as dynamic scaling?

In theory, you can run Cloud Foundry on servers nested inside a Faraday cage in your basement. You would have a totally isolated private cloud. The opposite side of the spectrum is paying for Cloud Foundry, as a service. Since Cloud Foundry is IaaS-neutral, i.e. it doesn't care whether you run it on Amazon or some other infrastructure. Somewhere in the middle of this spectrum is running it on shared hosting, shared clouds, private clouds and hybrid clouds.

If you, for instance, need to be HIPAA (Health Insurance Portability and Accountability Act) or PCI DSS (Payment Card Industry Data Security Standard) compliant, complete isolation and privacy is required. But if you are writing a social-networking web application, running in a shared cloud environment is perfectly reasonable.

At this point, you will know if you need to be private or not-private. Now you will need to choose a provider that supports the set of languages that you want. For instance, currently only ActiveState Stackato support Perl 5, so if you want to use Perl, your current only option is to run your own cloud or use Stackato. On the other hand, both AppFog and Paas.io support PHP, so you could choose between either. New Cloud Foundry based startups and forks are being announced almost weekly, so the options are constantly changing.

The last question to answer is the need for value-add services such as auto-scaling. Auto-scaling is the process of the application being monitored and dynamically scaled as needed. For instance, usually your web app is fine with a single PostgreSQL database and a single web server. But then you get Slashdot'ed! (The author does not read Reddit). Auto-scaling would spin up as many slave read-only database servers, load balancers and web servers to handle the tsunami of traffic. When the traffic spike is over, auto-scaling reduces the number of servers to the needed amount. This process, if it works right, can greatly reduce costs, since you only need to pay for virtualized hardware when you actually need it. If you had to own the physical hardware to support your highest traffic spike, your resources would sit around unused most of the time.

Installation

Simplest

ActiveState Stackato provides the easiest way to get a working Cloud Foundry instance running locally. There is a single, simple, beautiful command which takes care of everything, including creating a pristine VM:

```
curl get.stackato.com/microcloud | bash
```

This should pop up a VM running inside virtualbox which has Stackato running! You can then connect to the web interface at TODO.

From Github

On a 64bit Ubuntu LTS (10.04.2 works well) with at least 1GB RAM

```
sudo apt-get install openssh-server curl
bash < <(curl -s -k -B http://git.io/vcap_dev_setup)
```

But alas, vcap_dev_setup does not actually start your local Cloud Foundry instance, to do that you must:

```
~/cloudfoundry/vcap/dev_setup/bin/vcap_dev start
```

To see the status of your local instance:

```
~/cloudfoundry/vcap/dev_setup/bin/vcap_dev status
```

To target vmc at your newly started local CF instance type:

```
vmc target api.vcap.me
```

Deploying Cloud Foundry Instances

There is very good documenation in the the Cloud Foundry oss-docs repository about single and multi node vcap deployments, so I will not repeat that here. There is also a very newly released tool called BOSH which can be used to manage one or many Cloud Foundry installations. But alas, this tool needs a book itself. Keep an eye out for the Little BOSH Book.

How To Contribute

Many individuals and companies have contributed to Cloud Foundry to make it what it is today. Things that were added by the community include Python, PHP, Rack, JRuby, Erlang and many other features and services.

In the past, VMware was overwhelmed with Github pull requests, so some of them went seemingly ignored. This is explained by the fact that it was VMware's practice to sync from their private Gerrit repositories to their public Github mirror roughly monthly which greatly increases the likelihood of merge conflicts. Many pull requests were greatfully merged, but the waves of code would not stop.

To the delight of many, a public Gerrit instance was recently announced. This allows internal VMware CF developers and external CF developers to work together, in public, which is a huge step in the right direction for VMware. http://reviews.cloudfoundry.org Developers can sign in with OpenID or a Google Account and participate in the development of Cloud Foundry.

The following diagram is a rough outline of how the Cloud Foundry open source community works:

Hacking on Cloud Foundry

First, I highly encourage you to install the extremely handy Gerrit command-line gem :

```
gem install gerrit-cli
```

You may need sudo if you are installing into your system gem location. Now, you should have a gerrit binary in your PATH and so you can use it to clone the central Cloud Foundry Git repository, ``vcap'' :

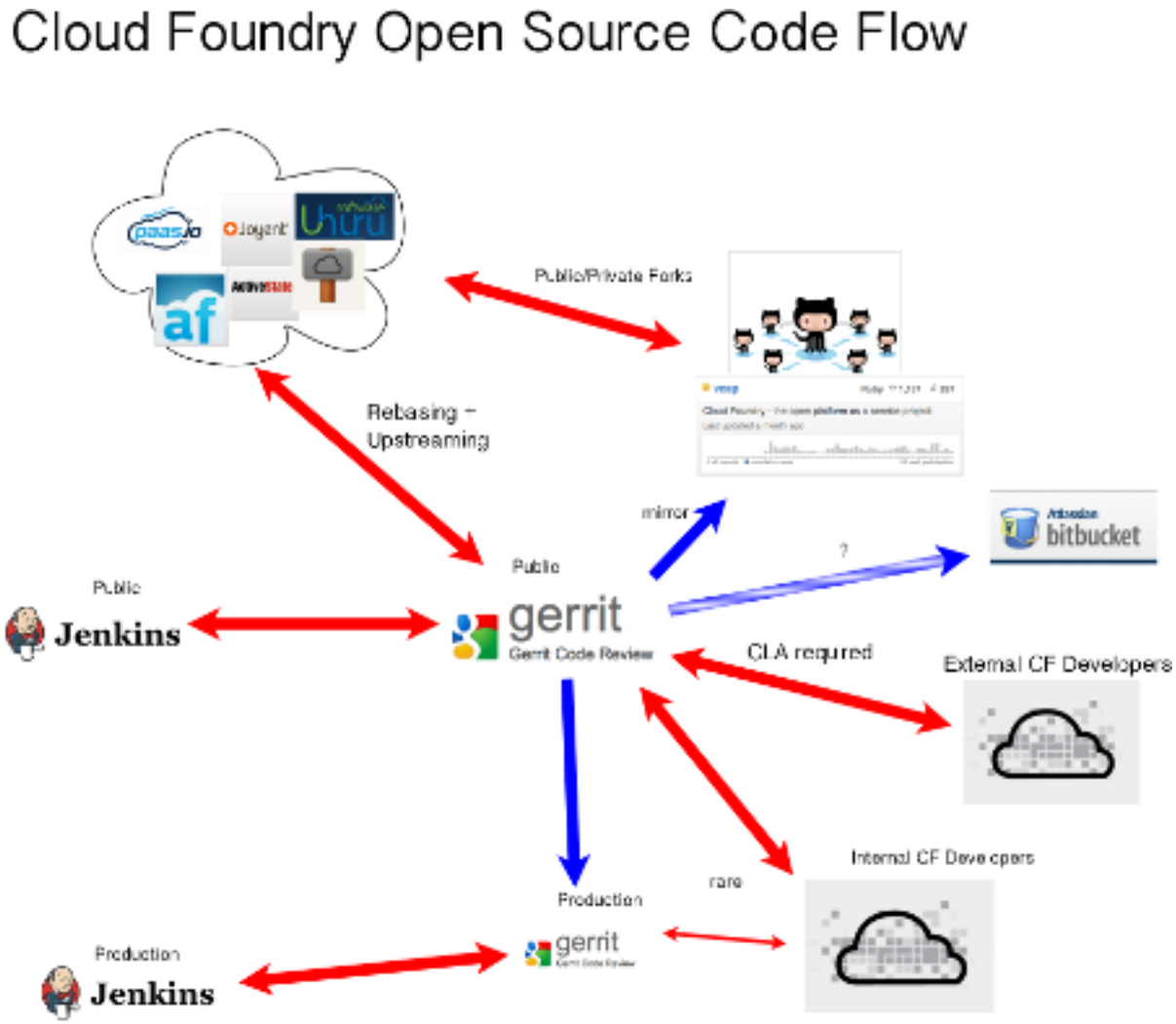

Figure 1: Cloud Foundry Community Process

```
# USERNAME is your Gerrit username, which could be different
# than your local username
gerrit clone ssh://USERNAME@reviews.cloudfoundry.org:29418/vcap
# make some coffee, enjoy it, come back

# If you have RVM, it will ask if you trust the .rvmrc
cd vcap

git submodule update --init --recursive
```

The last command above tells Git to recursively clone all the dependent repositories and ``mount'' them at the proper place in the directory structure. These dependent repos are vcap-services, vcap-tests, vcap-test-assets, vcap-java, acm and uaa. vcap-test-assets is a nested Git submodule, inside of vcap-tests.

Submitting Pull Requests

Cloud Foundry does not accept pull requests on Github, it is used purely as a mirror, for ``Github cruisers''. All contributions need to be made on the public Gerrit at `http://reviews.cloudfoundry.org`.

For a more detailed explanation of the contribution process, read the Open Source Developer workflow

Running Tests

All tests in Cloud Foundry are run with the Ruby `rake` tool.

```
rake spec          # Run specs
rake spec:rcov     # Run specs using RCov
rake tests         # Run integration tests.
```

Getting Help In Realtime-ish

There is a Cloud Foundry IRC channel on Freenode at #cloudfoundry. If you need help with Gerrit, there is also a very friendly community in #gerrit to help you.

App Portability

``App portability'' is the concept that your application should be not locked into a single vendor. Your application should be portable in the sense that you can take it and run it on any PaaS provider of your choosing.

To help this, Ingy has created a curated app gallery at https://github.com/Cloud-Apps . The goal is to fork projects into this Github organization and then keep track of the changes necessary to get it to run on each Paas.

Glossary

ACM

Access Control Manager. A general system for implementing access control features for applications.

BOSH

BOSH Outer SHell is an abstraction layer, between the Iaas and the Paas. It has a pluggable architecture and currently supports Amazon and VMWare vCloud.

caldecott

A Ruby gem which allows you to ``tunnel'' into the various services attached to your application. As an example, you can use caldecott to connect to the mysql command-line of your application running in the cloud with a command similar to

```
vmc tunnel mysql-12345
```

cloud controller

The Cloud Controller can be thought of as the maestro or orchestrator. It also includes the endpoints which vmc communicates with.

chef

Chef is an open-source systems integration framework to automate cloud-related tasks.

DEA

Droplet Execution Agent. A process which manages the running application instances.

dev-ops

The boundary layer between development and operations is referred to as ``dev-ops''. It is characterized by the best practices of operations and development coming together, TODO: EXAMPLE.

droplet

An application, along with all dependencies, in a compressed archive file.

gerrit

A code review tool based on Git. Gerrit was originally written for the Android Open Source Project at Google and allows many kinds of complex requirements and business rules to be taken into account when reviewing code.

health
Applications can exhibit different states of ``health'' which include:

- running
- stopped
- flapping
- restarting

health manager
The subsystem of CF which monitors the health of applications and classifies them into a certain number of ``states'', such as ``healthy'' or ``stopped''.

IAAS
Infrastructure-as-a-service. This is the layer below Platform-as-a-service. An example is Amazon EC2, which takes care of the ``infrastructure'' of the actual physical hardware.

lxc
Linux ``containers'' which are similar to FreeBSD ``jails''. They allow many ``virtual'' instances of Linux to run inside of a ``host'' Linux, which is actually running on hardware. These have been available in the mainline Linux kernel since 2.6.29.

manifest
A Manifest is a YAML configuration file which lists various properties and metadata about the application, such as endpoint URLs, package versions and checksums.

PAAS
Platform-as-a-service. Examples of this are Cloud Foundry, OpenShift, Rackspace, Google App Engine and many more.

router
The Router takes HTTP requests and sends them to the appropriate running application instance. It usually sits behind one or more load balancers.

rvm
Ruby Version Manager. A utitity which makes it simple to manage many different versions of Ruby as well as isolating the dependencies on a per-project level.

I highly recommend using rvm to isolate the Ruby that Cloud Foundry wants (currently 1.9.2-p180), versus what your system Ruby is, which you should most likely leave alone. RVM solves exactly the same problem as the virtualenv tool solves in Python as well as the perlbrew utiltiy for Perl 5.

stager
The stager takes an application, which is often a directory of files on disk, and turns it into a package that can be deployed to an application instance as part of an update or initial deploy.

Ubuntu LTS
Long-term supported releases of Ubuntu are supported for 2 years by Canonical. Currently it is the 12.04, which means 10.04, a widely-deployed version of Ubuntu, is recently unsupported.

UAA
User Account and Authentication. UAA uses OpenID Connect for authentication (also known as Single Sign On) and the OAuth2 protocol for granting access to resources.

vcap
VMware's Cloud Application Platform. This is the central Git repository which contains the Cloud Foundry codebase.

vcloud
VMWare's IaaS which is comprised of various VMWare products together with a vCloud API.

vmc
Stands for ``VMware Cloud'' or ``VMware controller'' or whatever you want it to mean, really. It is a ruby gem which is the command-line client to endpoints which implement the Cloud Foundry API.

warden
Warden is the Cloud Foundry security and isolation subsystem. It manages Linux containers, including their creation, destruction and monitoring. It can be thought of as a delicious layer of Ruby on top of a bit of C which uses Linux kernel hooks to strictly enforce resource limits and manage virtualized Linux instances on a ``host'' Linux server.

Useful Links

`http://rewews.cloudfoundry.org` - Public Gerrit for CF Contributions

`http://cloudfoundry.org` - VMware Cloud Foundry community website

`https://github.com/cloudfoundry` - Cloud Foundry Github Organization

`https://github.com/cloudfoundry/oss-docs` Cloud Foundry OSS Docs

`http://apidocs.cloudfoundry.com` - Community API Docs

`http://github.com/Cloud-Apps` - A curated Gallery of web apps that run on one or more PaaS's.