

A Running Tour of Cloud Foundry

Josh Long, Spring Developer Advocate
SpringSource, a division of VMware

Twitter: @starbuxman
Email: josh.long@springsource.com

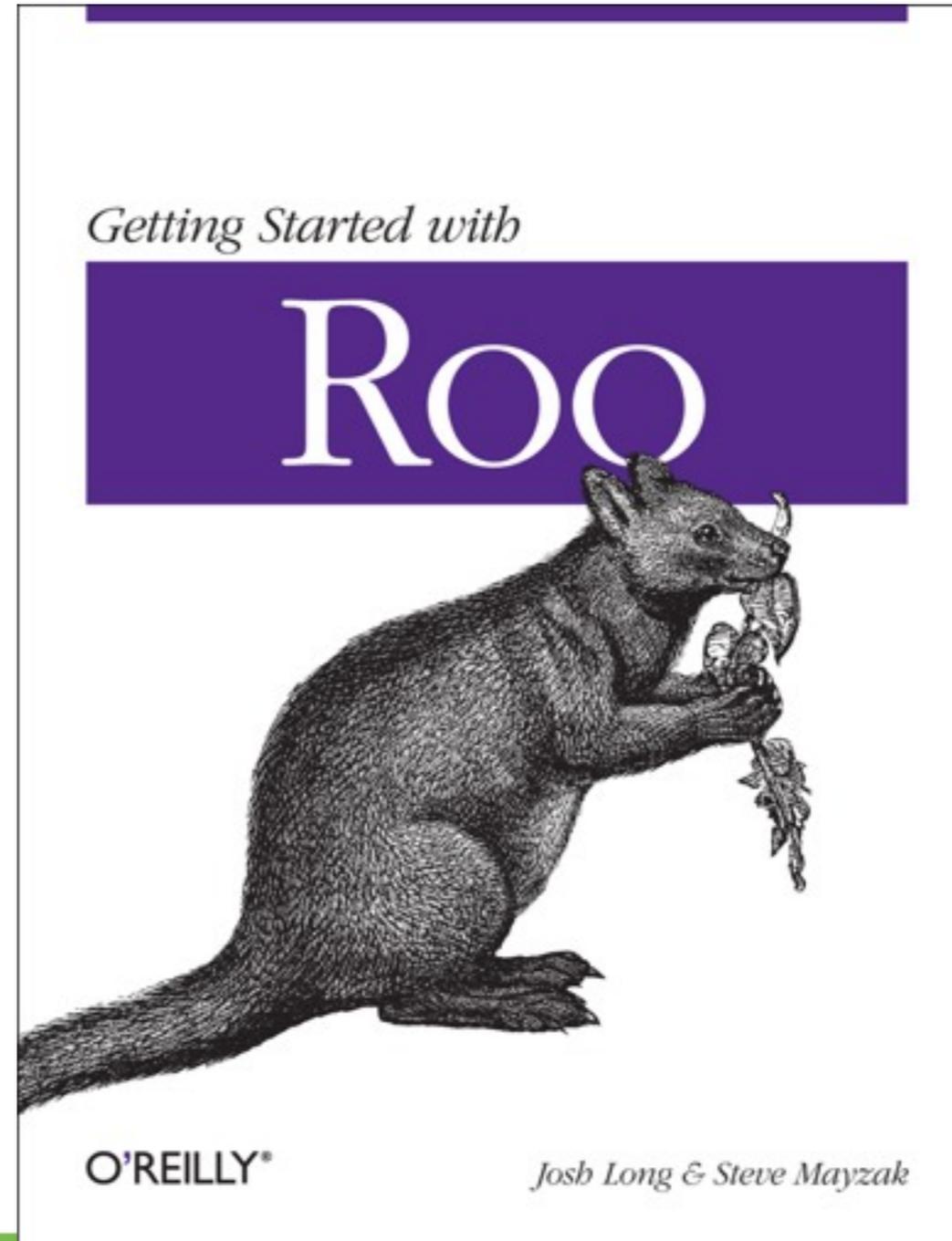
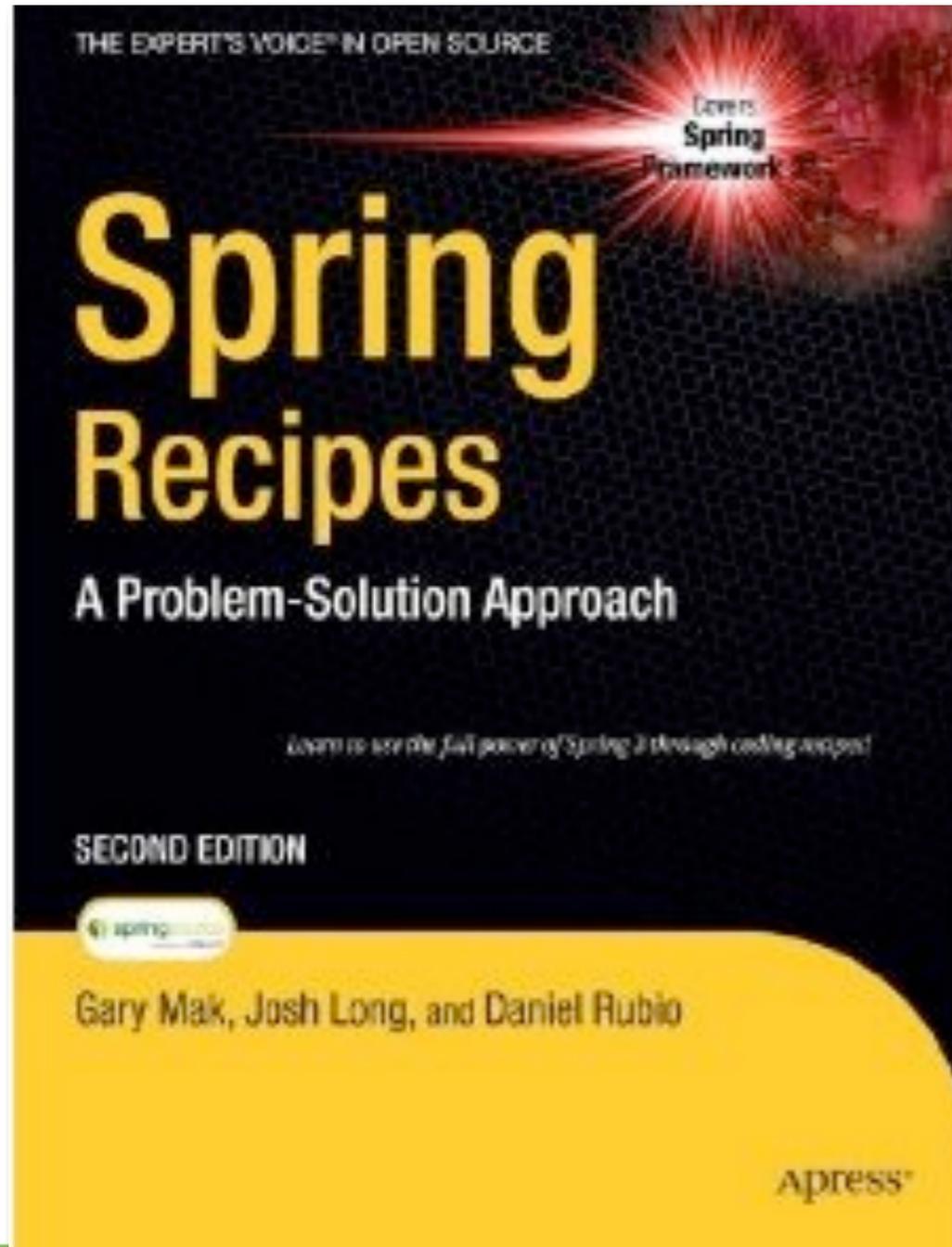


About Josh Long

Spring Developer Advocate

twitter: @starbuxman

josh.long@springsource.com



Spring Makes Building Applications Easy...

Tell Spring About Your Objects

```
package the.package.with.beans.in.it;
```

```
@Service
```

```
public class CustomerService {
```

```
    public Customer createCustomer(String firstName,  
                                   String lastName,  
                                   Date signupDate) {
```

```
    }
```

```
    ...
```

```
}
```

I want Database Access ... with Hibernate 4 Support

```
package the.package.with.beans.in.it;
```

```
...
```

```
@Service
```

```
public class CustomerService {
```

```
    @javax.inject.Inject
```

```
    private SessionFactory sessionFactory;
```

```
    public Customer createCustomer(String firstName,  
                                   String lastName,  
                                   Date signupDate) {
```

```
        Customer customer = new Customer();
```

```
        customer.setFirstName(firstName);
```

```
        customer.setLastName(lastName);
```

```
        customer.setSignupDate(signupDate);
```

```
        sessionFactory.getCurrentSession().save(customer);
```

```
        return customer;
```

```
    }
```

```
...
```

I want Declarative Transaction Management...

```
package the.package.with.beans.in.it;
...
@Service
public class CustomerService {

    @javax.inject.Inject
    private SessionFactory sessionFactory;

    @Transactional
    public Customer createCustomer(String firstName,
                                   String lastName,
                                   Date signupDate) {

        Customer customer = new Customer();
        customer.setFirstName(firstName);
        customer.setLastName(lastName);
        customer.setSignupDate(signupDate);

        sessionFactory.getCurrentSession().save(customer);
        return customer;
    }
    ...
}
```

I want Declarative Cache Management...

```
package the.package.with.beans.in.it;
...
@Service
public class CustomerService {

    @javax.inject.Inject
    private SessionFactory sessionFactory;

    @Transactional
    @Cacheable("customers")
    public Customer createCustomer(String firstName,
                                   String lastName,
                                   Date signupDate) {

        Customer customer = new Customer();
        customer.setFirstName(firstName);
        customer.setLastName(lastName);
        customer.setSignupDate(signupDate);

        sessionFactory.getCurrentSession().save(customer);
        return customer;
    }
}
```

I want a RESTful Endpoint...

```
package org.springframework.examples.spring31.web;
```

```
..
```

```
@Controller
```

```
public class CustomerController {
```

```
    @Autowired
```

```
    private CustomerService customerService;
```

```
    @RequestMapping(value = "/customer/{id}",  
                    produces = MediaType.APPLICATION_JSON_VALUE)
```

```
    @ResponseBody
```

```
    public Customer customerById(@PathVariable("id") Integer id) {
```

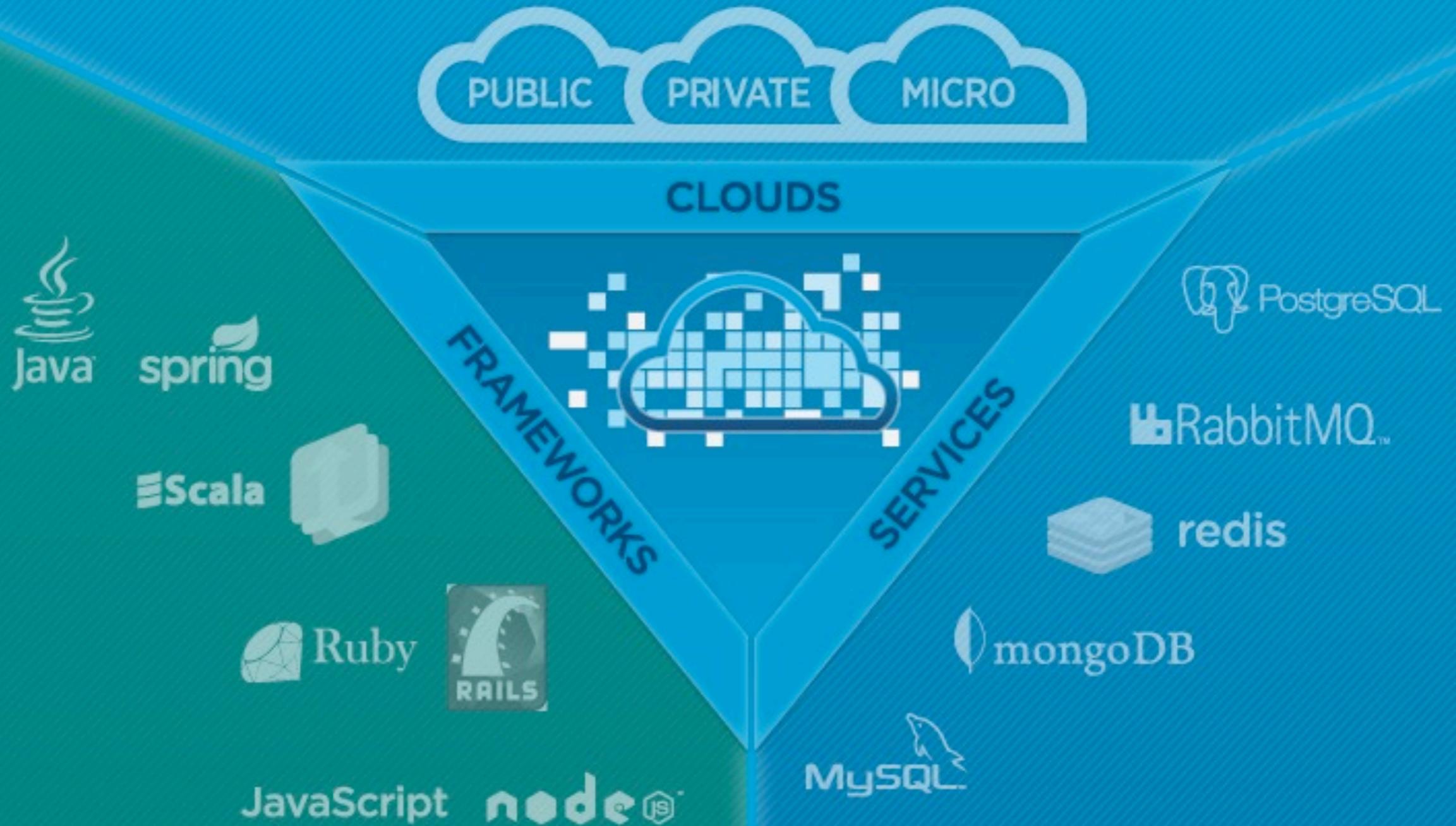
```
        return customerService.getCustomerById(id);
```

```
    }
```

```
    ...
```

```
}
```

Cloud Foundry: Choice of Runtimes



Frameworks and Runtimes Supported

- **Out of the Box**

- **Java** (.WAR files, on Tomcat. Spring's an ideal choice here, of course..)
- **Scala** (Lift, Play!)
- **Ruby** (Rails, Sinatra, etc.)
- **Node.js**

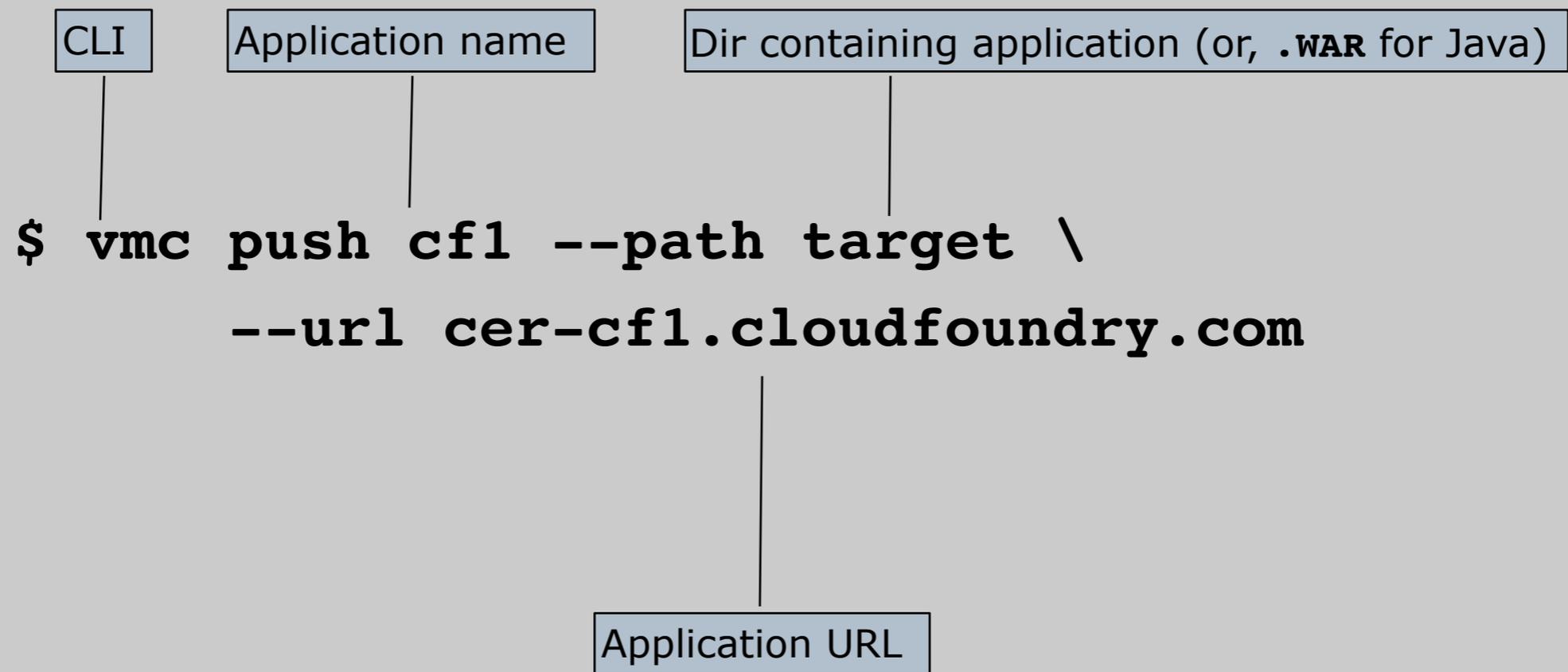
- **Other**

- **Python (Stackato)**
- **PHP (AppFog)**
- **Haskell (1)**
- **Erlang (2)**

1) <http://www.cakesolutions.net/teamblogs/2011/11/25/haskell-happstack-on-cloudfoundry/>

2) <https://github.com/cloudfoundry/vcap/pull/20>

Deploying an Application



Deploying an Application

```
$ vmc push cf1 --path target \  
    --url cer-cf1.cloudfoundry.com
```

```
Detected a Java Web Application, is this correct?  
[Yn]:
```

Deploying an Application

```
$ vmc push cf1 --path target \  
    --url cer-cf1.cloudfoundry.com
```

```
Detected a Java Web Application, is this correct?  
[Yn]:
```

```
Memory Reservation [Default:512M] (64M, 128M, 256M,  
512M, 1G or 2G)
```

Deploying an Application

```
$ vmc push cf1 --path target \  
    --url cer-cf1.cloudfoundry.com
```

```
Detected a Java Web Application, is this correct?  
[Yn]:
```

```
Memory Reservation [Default:512M] (64M, 128M, 256M,  
512M, 1G or 2G)
```

```
Creating Application: OK
```

```
Would you like to bind any services to 'cf1'? [yN]:
```

Deploying an Application

```
$ vmc push cf1 --path target \  
    --url cer-cf1.cloudfoundry.com
```

```
Detected a Java Web Application, is this correct?  
[Yn]:
```

```
Memory Reservation [Default:512M] (64M, 128M, 256M,  
512M, 1G or 2G)
```

```
Creating Application: OK
```

```
Would you like to bind any services to 'cf1'? [yN]:
```

```
Uploading Application:
```

```
  Checking for available resources: OK
```

```
  Packing application: OK
```

```
  Uploading (2K): OK
```

```
Push Status: OK
```

```
Starting Application: OK
```

Deploying an Application (with a Manifest)

```
$ vmc push
```

```
Would you like to deploy from the current directory?
```

```
[Yn]: y
```

```
Pushing application 'html5expenses'...
```

```
Creating Application: OK
```

```
Creating Service [expenses-mongo]: OK
```

```
Binding Service [expenses-mongo]: OK
```

```
Creating Service [expenses-postgresql]: OK
```

```
Binding Service [expenses-postgresql]: OK
```

```
Uploading Application:
```

```
  Checking for available resources: OK
```

```
  Processing resources: OK
```

```
  Packing application: OK
```

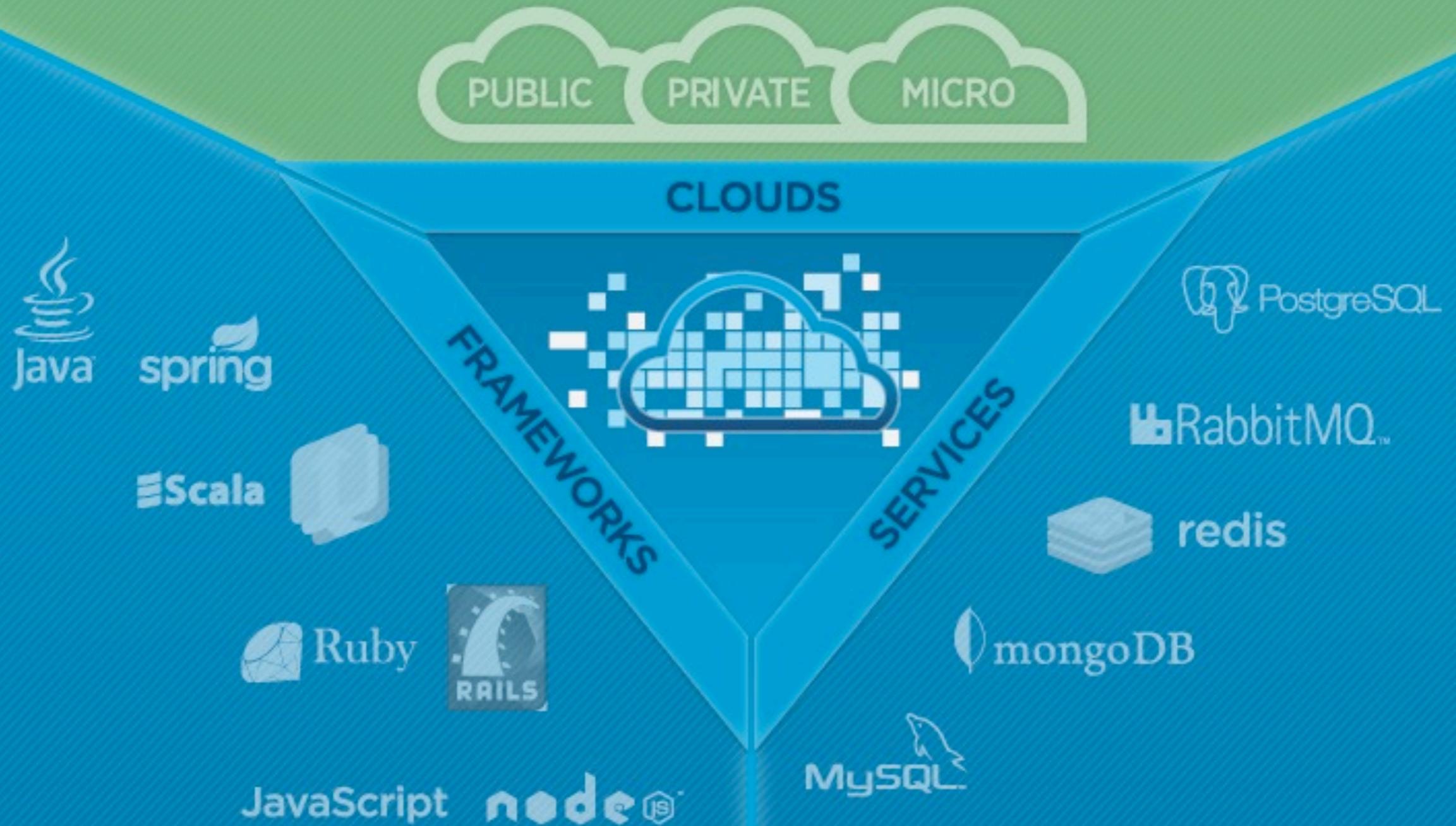
```
  Uploading (6K): OK
```

```
Push Status: OK
```

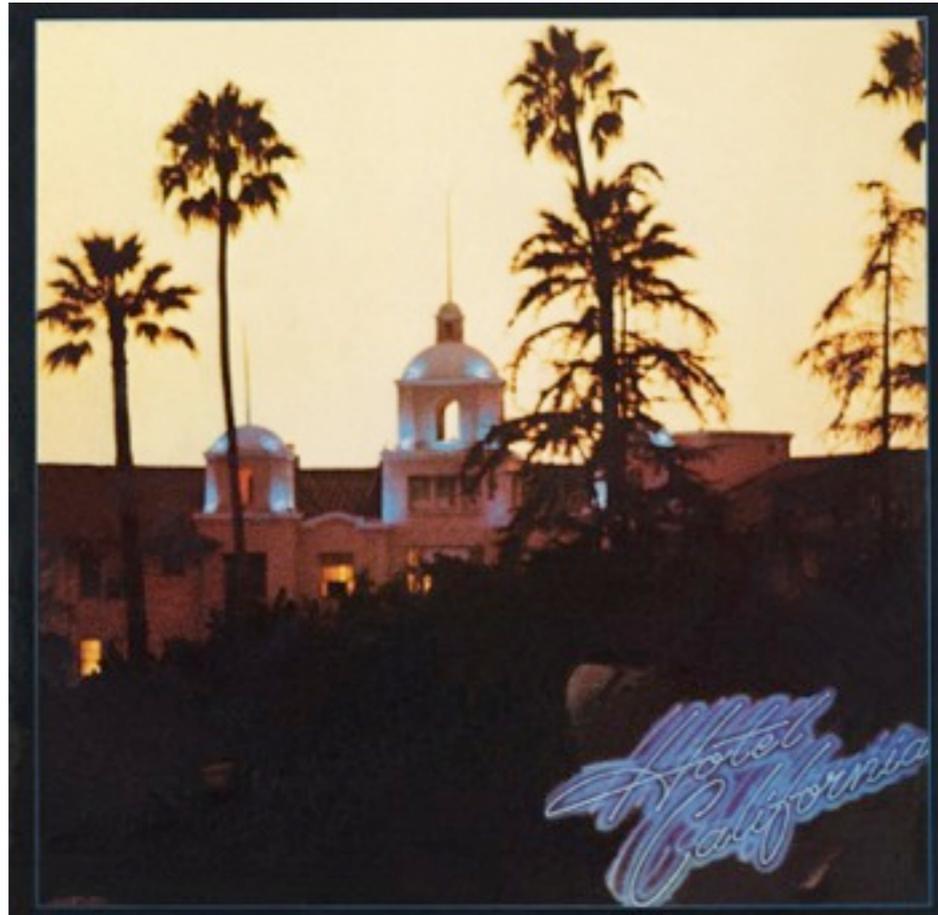
Deploying an Application (with a `manifest.yml`)

```
---
applications:
  target:
    name: html5expenses
    url: ${name}.${target-base}
    framework:
      name: spring
      info:
        mem: 512M
        description: Java SpringSource Spring Application
        exec:
          mem: 512M
          instances: 1
      services:
        expenses-mongo:
          type: :mongodb
        expenses-postgresql:
          type: :postgresql
```

Cloud Foundry: Choice of Clouds



Main Risk: Lock In

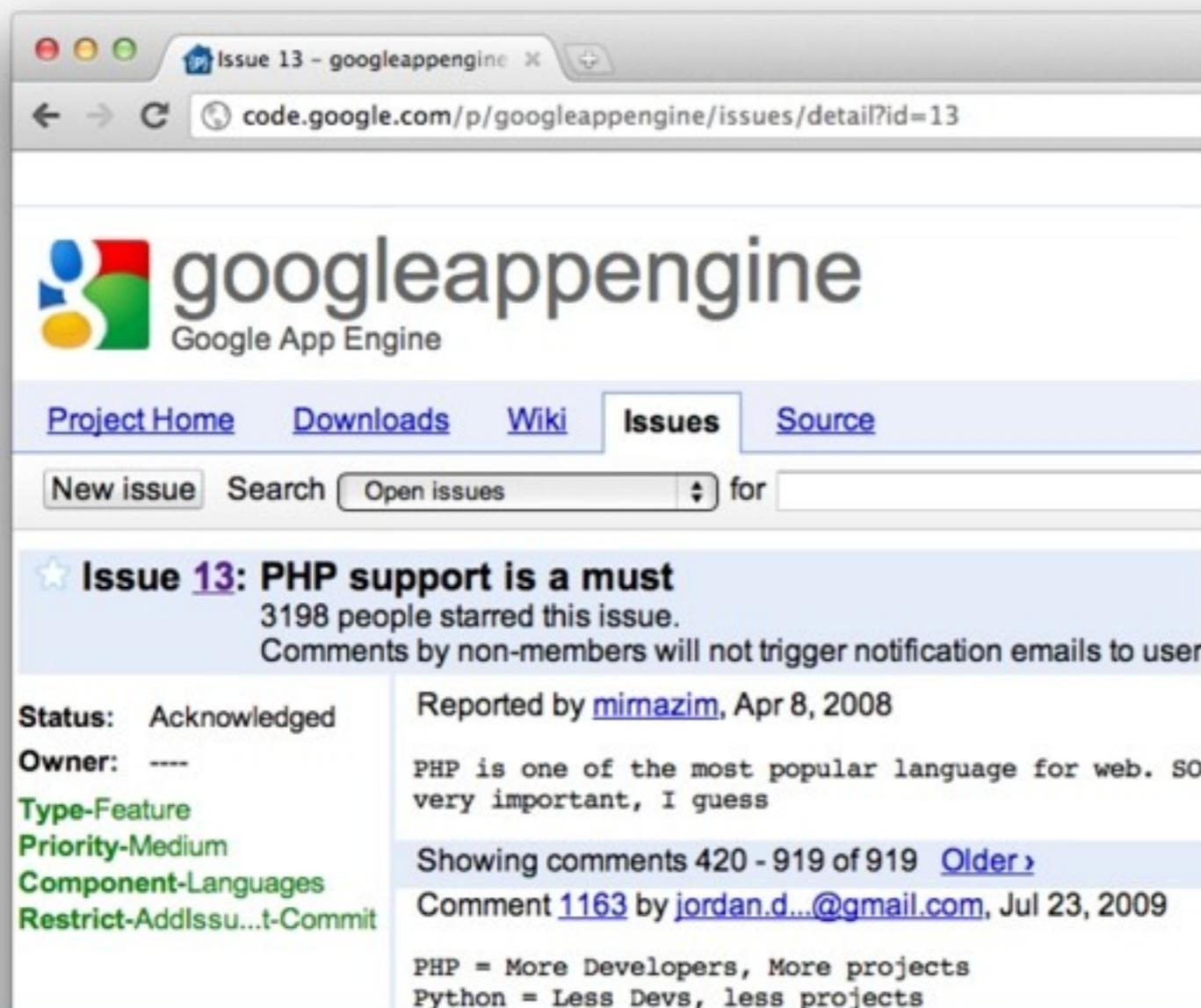


*Welcome to the hotel california
Such a lovely place
Such a lovely face
Plenty of room at the hotel california
Any time of year, you can find it here*

*Last thing I remember, I was
Running for the door
I had to find the passage back
To the place I was before
'relax,' said the night man,
We are programmed to receive.
**You can checkout any time you like,
But you can never leave!***

-the Eagles

Open Source Advantage



The screenshot shows a web browser window with the URL `code.google.com/p/googleappengine/issues/detail?id=13`. The page header includes the Google App Engine logo and navigation links for Project Home, Downloads, Wiki, Issues, and Source. Below the header is a search bar with a 'New issue' button and a dropdown menu set to 'Open issues'. The main content area features a star icon next to the issue title 'Issue 13: PHP support is a must', followed by the text '3198 people starred this issue.' and a note that 'Comments by non-members will not trigger notification emails to users'. The issue details include a status of 'Acknowledged', an owner of '---', a type of 'Feature', a priority of 'Medium', and a component of 'Languages'. The issue was reported by 'mimazim' on April 8, 2008. The description reads: 'PHP is one of the most popular language for web. SO I think PHP support is very important, I guess'. There are two comments shown: comment 1163 by 'jordan.d...@gmail.com' on July 23, 2009, which says 'PHP = More Developers, More projects' and 'Python = Less Devs, less projects'; and comment 1666 by a project member 'i...@google.com' on January 6, 2011, which is highlighted in a blue box.

- added April 8, 2008
- 3198 people starred it
- eventually closed in January, 2011
- never added

Comment [1666](#) by project member [i...@google.com](#), Jan 6, 2011

I'm making this issue read-only. I think the points here have been made. There's no reason to email thousands of people every time someone says "+1".

There are no current plans to support PHP on App Engine. No one on this team is against the idea, and given unlimited resources, we would do it. At this time, bringing another language runtime to App Engine is unfeasible given the other goals we are trying to meet.

please support php.

Comment [1167](#) by [tapsboy](#), Jul 28, 2009

Open Source Advantage

The screenshot shows a web browser window with the URL <https://github.com/cloudfoundry/vcap/pull/25>. The page header includes the GitHub logo and the repository name 'cloudfoundry / vcap'. Below the header, there are tabs for 'Code', 'Network', and 'Pull Requests'. A red 'Closed' button is visible next to the pull request title 'paulj wants someone to merge 6 commits into cloudfoundry:mast'. Below the tabs, there are links for 'Discussion', 'Commits <> 6', and 'Diff >= 8'.

- Cloud Foundry announced April, 2011
- Community PHP support submitted April, 2011
- Code pulled in August, 2011



paulj opened this pull request April 17, 2011

PHP support

Support for PHP based applications.

- Uses lighttpd as a front-end onto a php-fastcgi worker
- Includes minor changes to the common.rb staging support to allow stop scripts to be overridden
- Includes a php.md documentation file describing how Wordpress would be installed using the patch
- Requires an equivalent patch in vmc, raised as <https://github.com/cloudfoundry/vmc/pull/4>



paulj, pbozeman, olegshaldybin, and davidstrauss are participating in this pull request.



+ paulj added some commits

April 15, 2011

d8a8cdb Initial PHP support via lighttpd.

803a14d Cleaning launching, shutdown, docs.



pbozeman commented

April 17, 2011

We'll get reviewed and pulled shortly, pending any feedback. That being said, before we can pull it, we want to add the ability to enable/disable frameworks so that we don't disrupt the large production system at Cloud Foundry until this has been through our load/stress tests. We'll enable it in the Microcloud context out of the gate though.

Closed

+ 198 additions

- 5 deletions

[All Pull Requests](#)

Cloud Foundry: Clouds



■ AppFog.com

- community lead for PHP
- PaaS for PHP



■ Joyent

- community lead for Node.js



■ ActiveState

- community lead for Python, Perl
- Providers of Stackato private PaaS

Cloud Foundry Community



cloudfoundry (The open p

Name	The open platform-as-a-service project
Email	support@cloudfoundry.org
Website/Blog	http://www.cloudfoundry.org
Member Since	Feb 16, 2011

vCENTER / vSPHERE

Public Repositories (7)

GITHUB



ENVIRONMENT



SCRIPTS



DEPLOY



vcap-services

Ruby 250

Cloud Foundry - the open platform as a service project

Last updated about 9 hours ago



Micro Cloud Foundry

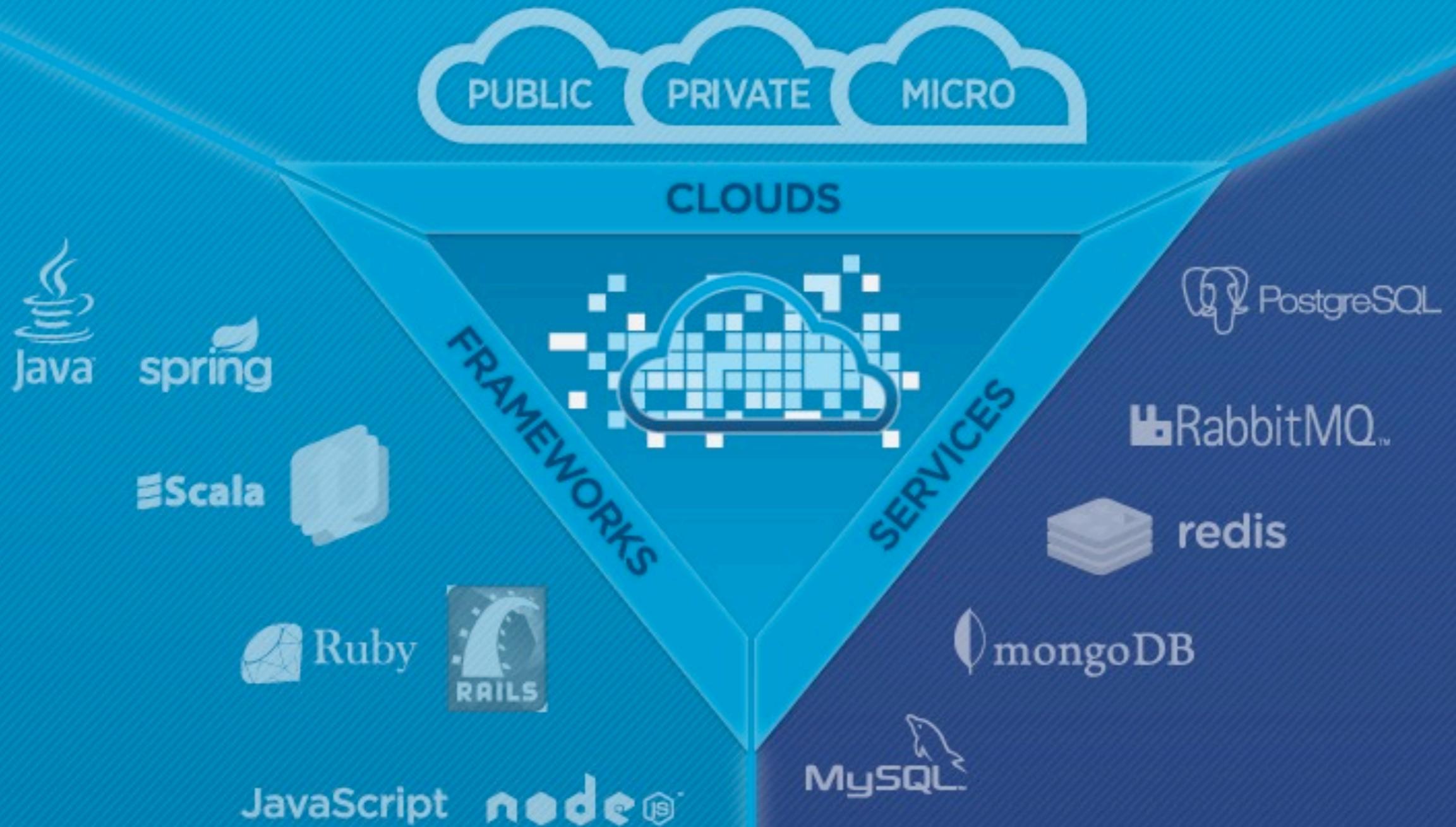
- **micro.cloudfoundry.com**
- **works on OS X, Windows, Linux**



The screenshot shows a web browser window with the URL <https://my.cloudfoundry.com/micro>. The page features a "BETA" badge in the top left corner. The main heading is "Micro Cloud Foundry" with a sub-heading "Micro Cloud Foundry™ - now you can run a complete instance of Cloud Foundry on your own computer." Below this, it states "Getting started is as simple as 1, 2, 3:" and lists three steps: 1. Install (VMware Player or VMware Workstation for Windows and Linux users; VMware Fusion for Mac OS X users), 2. Login (Using your CloudFoundry.com credentials*), and 3. Download (Your Micro Cloud Foundry virtual machine image). A "Learn more" section provides links to a screencast, a getting started guide, and a knowledge base. A login form on the right side includes fields for "Email" and "Password" and a "Login" button.



Cloud Foundry: Services



Cloud Foundry: Services

- **Services are one of the extensibility planes in Cloud Foundry**
 - there are more services being contributed by the community daily!
- **MySQL, Redis, MongoDB, RabbitMQ, PostgreSQL**
- **Services may be shared across applications**
- **Cloud Foundry abstracts the provisioning aspect of services through a uniform API hosted in the cloud controller**
- **It's very easy to take an app and add a service to the app in a uniform way**
 - Cassandra? COBOL / CICS, Oracle

Cloud Foundry: Services

```
$ vmc create-service mysql --name mysql1
```

```
Creating Service: OK
```

```
$ vmc services
```

```
==== System Services =====
```

Service	Version	Description
mongodb	1.8	MongoDB NoSQL store
mysql	5.1	MySQL database service
postgresql	9.0	PostgreSQL database service (vFabric)
rabbitmq	2.4	RabbitMQ messaging service
redis	2.2	Redis key-value store service

```
==== Provisioned Services =====
```

Name	Service
mysql1	mysql

Cloud Foundry: Services Creation and Binding

```
$VCAP_SERVICES:
{"redis-2.2":
  [{"name":"redis_sample","label":"redis-2.2","plan":"free",
    "tags":["redis","redis-2.2","key-value","nosql"],
    "credentials":
      {"hostname":"172.30.48.40",
        "host":"172.30.48.40",
        "port":5023,
        "password":"8e9a901f-987d-4544-9a9e-ab0c143b5142",
        "name":"de82c4bb-bd08-46c0-a850-af6534f71ca3"}
    ]},
  {"name":"mongodb-1.8","label":"mongodb-1.8","plan":"free","tags":.....}
```

Accessing Your Services

■ Debugging and accessing the data locally

- Caldecott --> Service tunneling. Access your Cloud Foundry service as if it was local.



Tunneling

```
gem install caldecott
```

```
vmc tunnel <mongodb>
```

```
Installing RDoc documentation for caldecott-0.0.4...
moni-air:developers_cloudfoundry ciberch$ vmc tunnel mongodb-92914
Deploying tunnel application 'caldecott'.
Create a password: *****
Uploading Application:
  Checking for available resources: OK
  Packing application: OK
  Uploading (1K): OK
Push Status: OK
Binding Service [mongodb-92914]: OK
Staging Application: OK
Starting Application: OK
Getting tunnel connection info: OK

Service connection info:
  username : 7344cf16-269e-4572-b1ff-c28f678bed34
  password : c383adb4-c4b8-446e-85bb-8d68278b0737
  name     : db

Starting tunnel to mongodb-92914 on port 10000.
1: none
2: mongo
Which client would you like to start?: 1
Open another shell to run command-line clients or
use a UI tool to connect using the displayed information.
Press Ctrl-C to exit...
```

Using your favorite tools

The image shows a screenshot of the MongoHub application interface. The main window, titled "MongoHub", displays a "localhost" connection icon (a stack of silver disks) and a terminal window at the bottom. The terminal window shows the following output:

```
Installing RDoc document
mon-i-air:developers_cl
Deploying tunnel appli
Create a password: ***
Uploading Application:
  Checking for availab
Packing application:
  Uploading (17%): 0%
```

Overlaid on the main window is a dialog box titled "Add New Connection". The dialog box contains the following fields and options:

- Alias: `mongodb-92914`
- Host: `localhost`
- Port: `10000`
- User: `7344cf16-269...`
- Passwd: `.....`
- DB: `db`
- Use Replica Set:
- Servers: `host1:port1,host2:port2,host3:p...`
- Set Name: `demo_repl`
- Use SSH Tunnel:

DATABASES

- db 4

DB

- system.users
- system.indexes
- git_hub_reposito...
- cloud_foundry_...

Collection db.cloud_foundry_app_in Stat Monitor Reconnect

Name	Value	Type
avgObjSize	656.000000	Double
count	2	Int
flags	1	Int
indexSizes		Object
lastExtentSize	8192	Int
nindexes	1	Int
ns	db.cloud_foundry_app_infos	String
numExtents	1	Int
ok	1.000000	Double
paddingFactor	1.000000	Double
size	1312	Int
storageSize	8192	Int
totalIndexSize	8192	Int

db.cloud_foundry_app_infos

Sort

Skip Limit Run

Name	Value	Type
app_urls		Array
description	The Box sample app has a redesigned interface for interacting with your content on Bo...	String
display_name	box-sample-ruby-app	String
env_vars		Object
framework	sinatra	String
instances	1	Int
memory	128	Int
repo_id	4f145ac56646652dd4000001	ObjectId
runtime	ruby19	String
starting_url	https://www.box.com/developers/services	String
thumb_url	/images/box-rebuilt-ruby/75.png	String
_id	4f145ac56646652dd4000001 Remove	ObjectId

Total Results: 2 (1.83s) Expand Collapse

code: <http://git.springsource.org/spring-samples/>



Questions?

Say hi on Twitter: [@starbuxman](https://twitter.com/starbuxman)