How AppFog Built a PaaS around CloudFoundry

Jeremy Voorhis
Senior Engineer, AppFog Inc
jeremy@appfog.com
@jvoorhis
http://www.appfog.com

Agenda

- What is PaaS?
- What is CloudFoundry?
- Adopting CloudFoundry at AppFog

What is PaaS?

"The capability [...] to deploy onto the cloud infrastructure [...] applications created using programming languages, libraries, services, and tools supported by the provider."

The NIST Definition of Cloud Computing

NIST 800-145

Why PaaS?

- Product teams focus on development
- Shortens the feedback loop
- Promotes horizontal scalability

A Word on NoOps

Follow Adrian

Adrian Cockcroft • Netflix runs NoOps - i.e. we do all the things that comments above say won't happen. Just because you don't think "enterprises" will stop using dedicated ops orgs, doesn't prevent some of us from running a dev-only product team that runs its own code directly and calling it NoOps. It's different, so it should have its own name. Get over it.

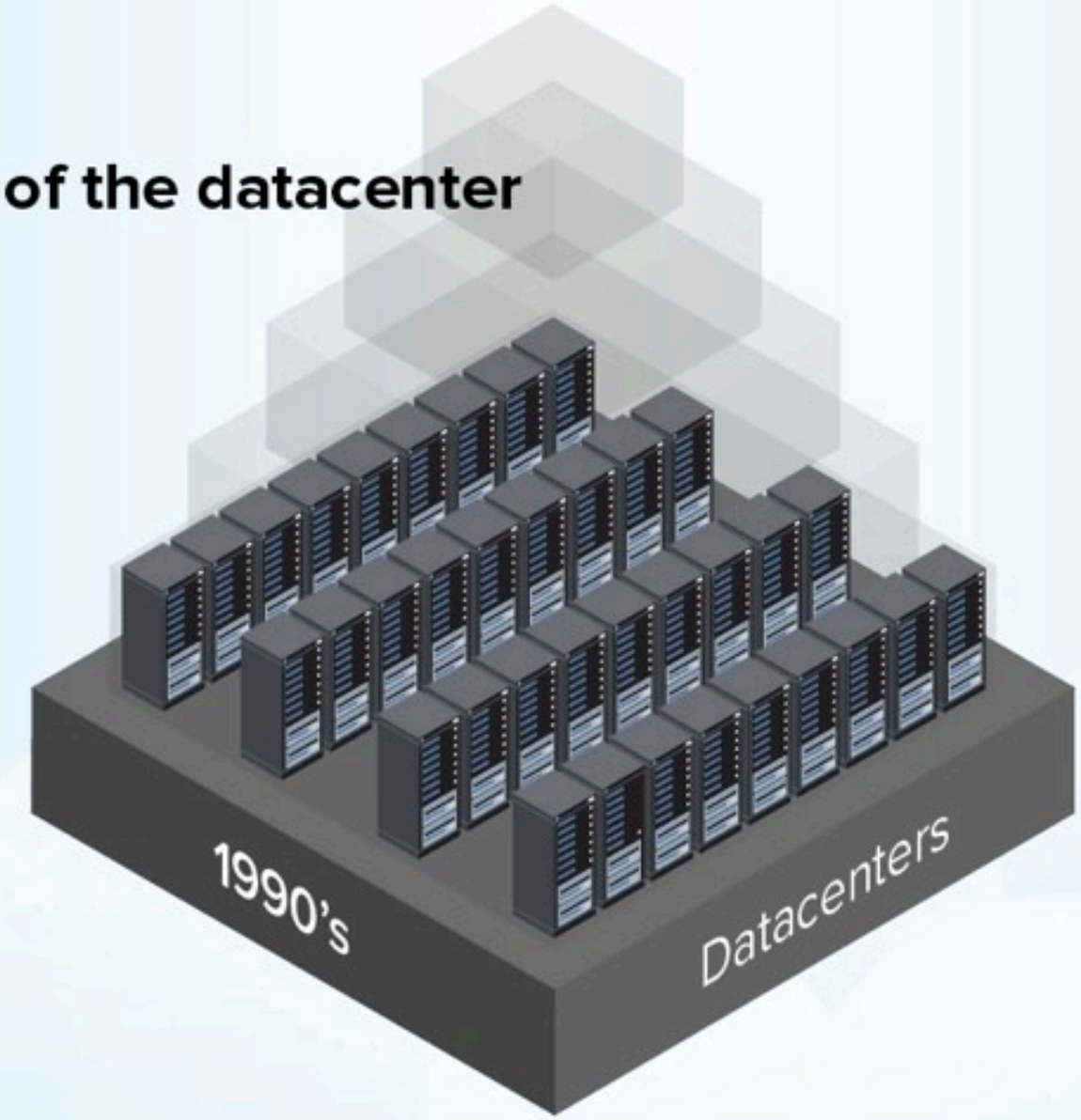
1 day ago • Like • Flag as inappropriate 5

http://www.linkedin.com/groups/Cloud-moves-towards-NoOps-world-4084799.S.92540468

1990s: The rise of the datacenter

Costs decrease

Sources: National Venture Capital Association and Center for Venture Research

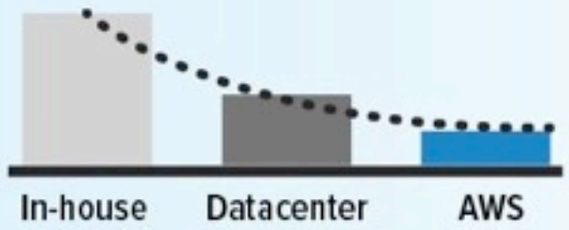


Productivity increases

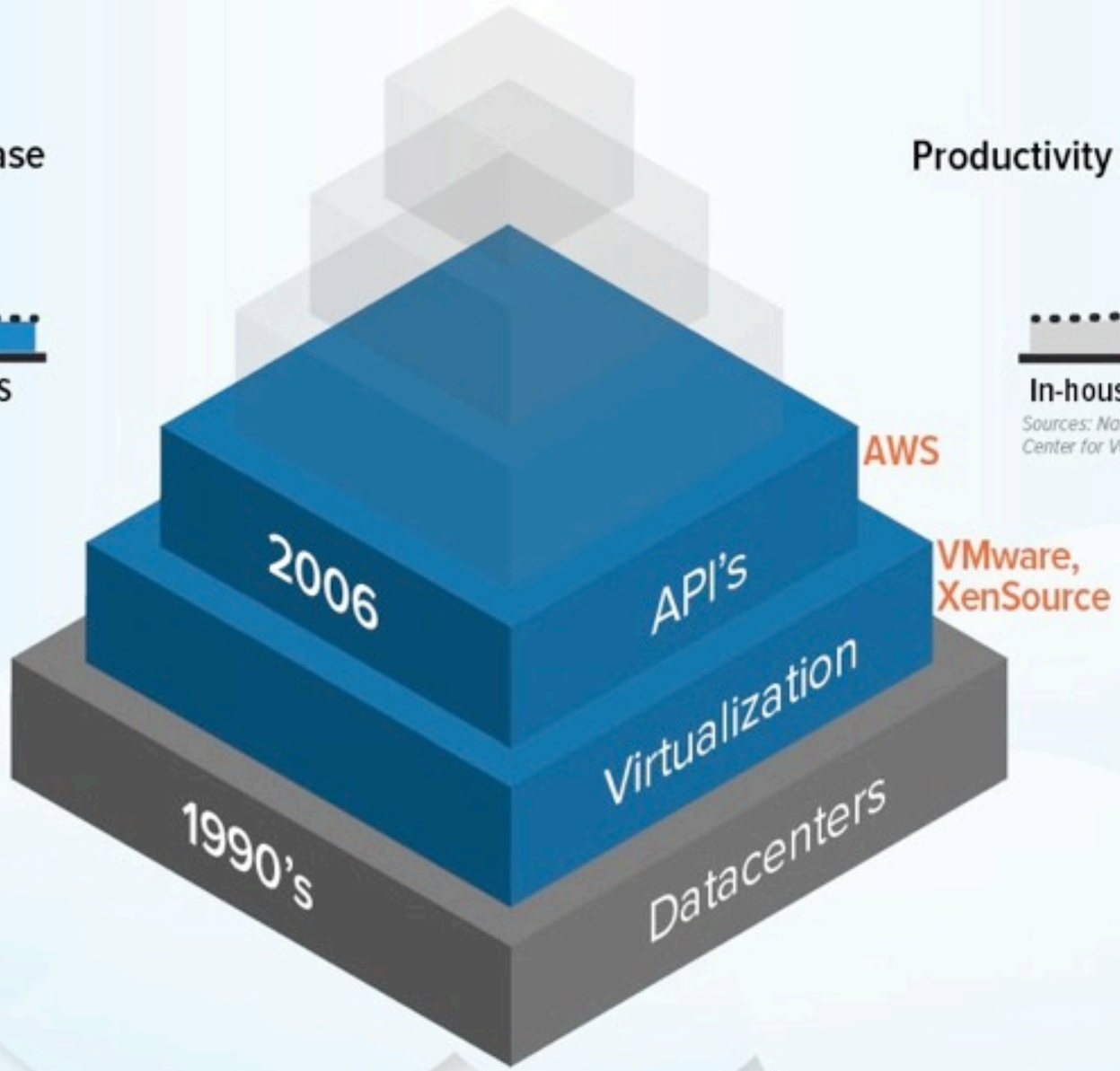
Sources: National Venture Capital Association and Center for Venture Research

Next: 21st century virtualization and Amazon Web Services

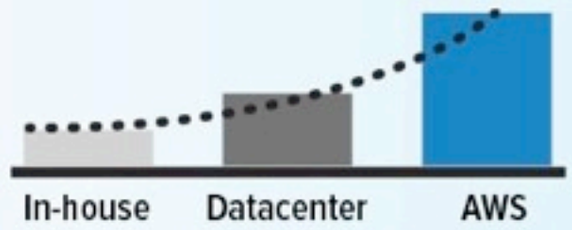
Costs exponentially decrease



Sources: National Venture Capital Association and Center for Venture Research



Productivity exponentially increases



Sources: National Venture Capital Association and Center for Venture Research

Data from the National Venture Capital Association and the Center for Venture Research

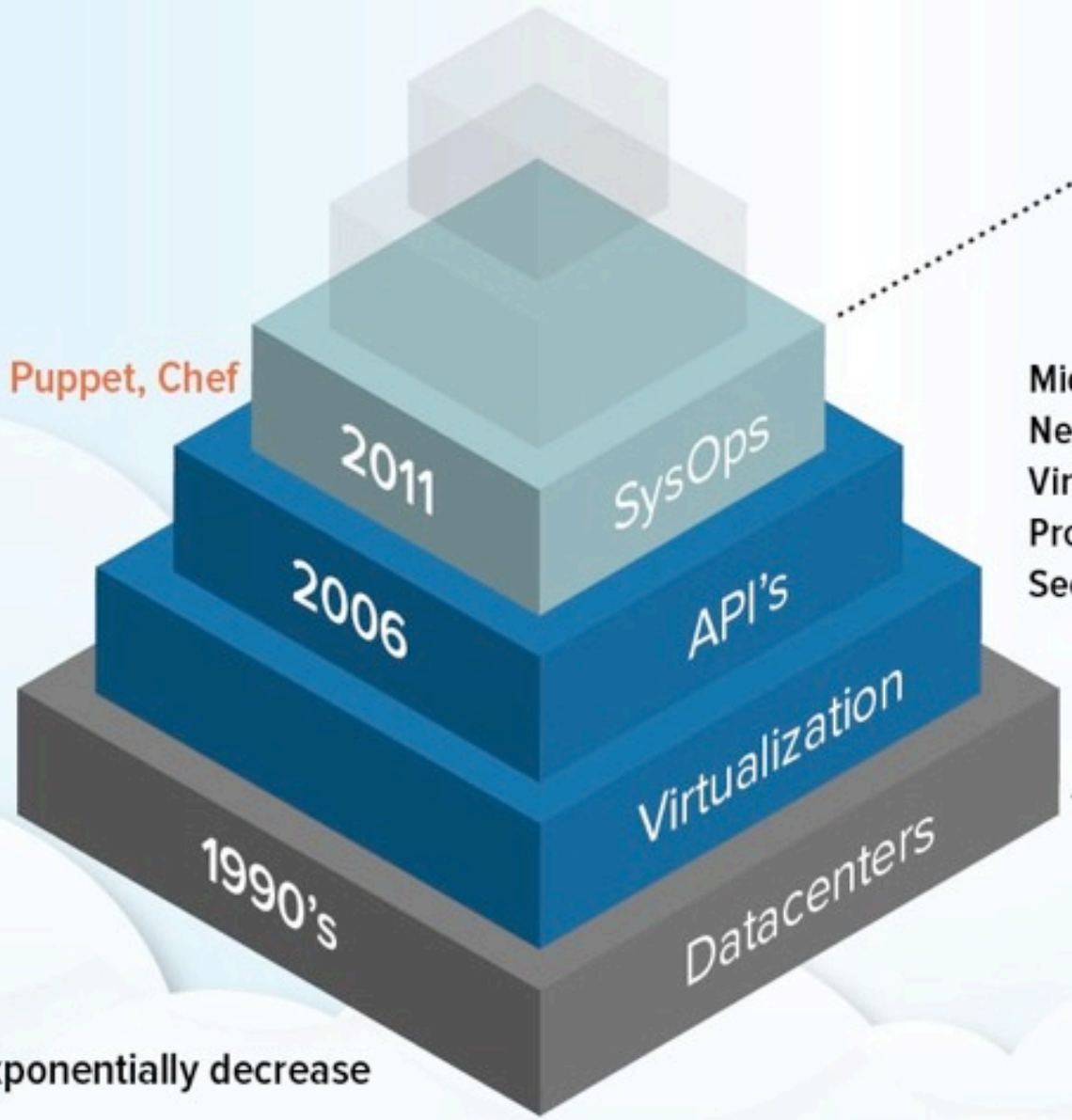
Year	Total Seed Deals	Total Seed Dollars (MM)
2001	48280	$30,689
2002	36178	$16,006
2003	42210	$18,426
2004	48222	$22,952
2005	49756	$24,020
2006	51391	$26,854
2007	57623	$27,613
2008	55998	$20,950
2009	57582	$19,349
2010	62286	$21,825

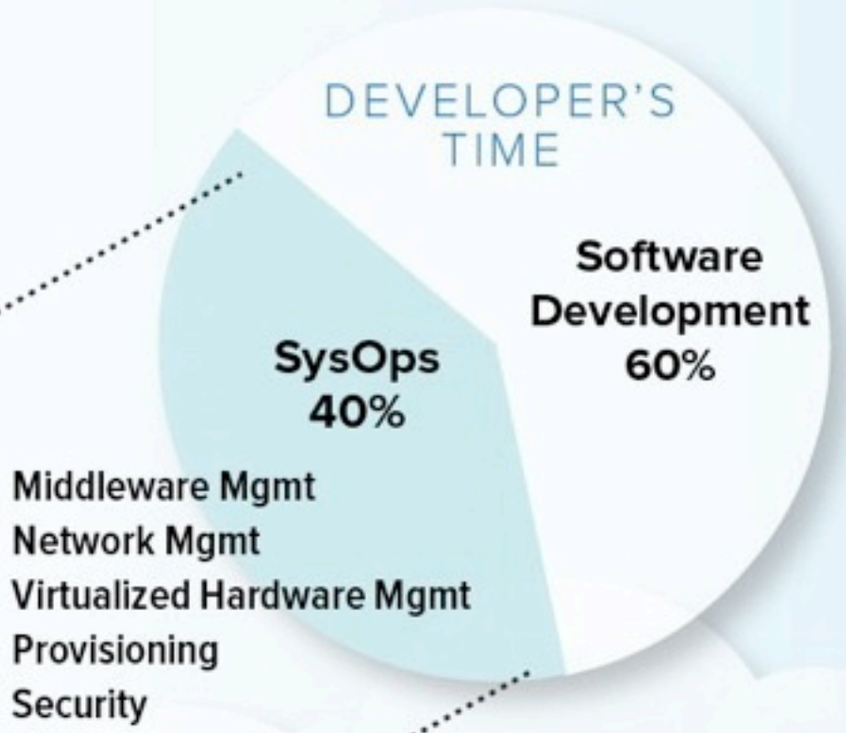
http://blog.appfog.com/appfog-entrepreneur-enabler-2/

Start-ups in the Early 2000's vs. 2010's

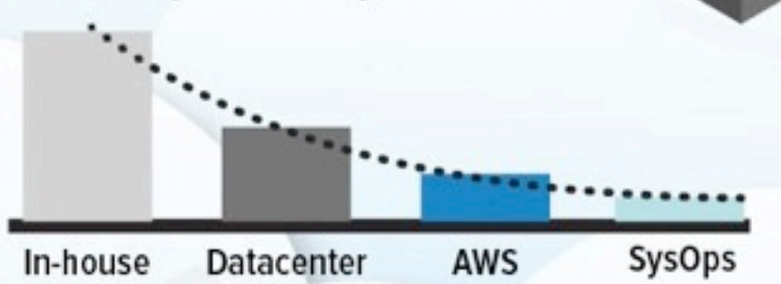
Source: National Venture Capital Association and the UNH Center for Venture Research

2011: Systematizing SysOps management





Costs exponentially decrease

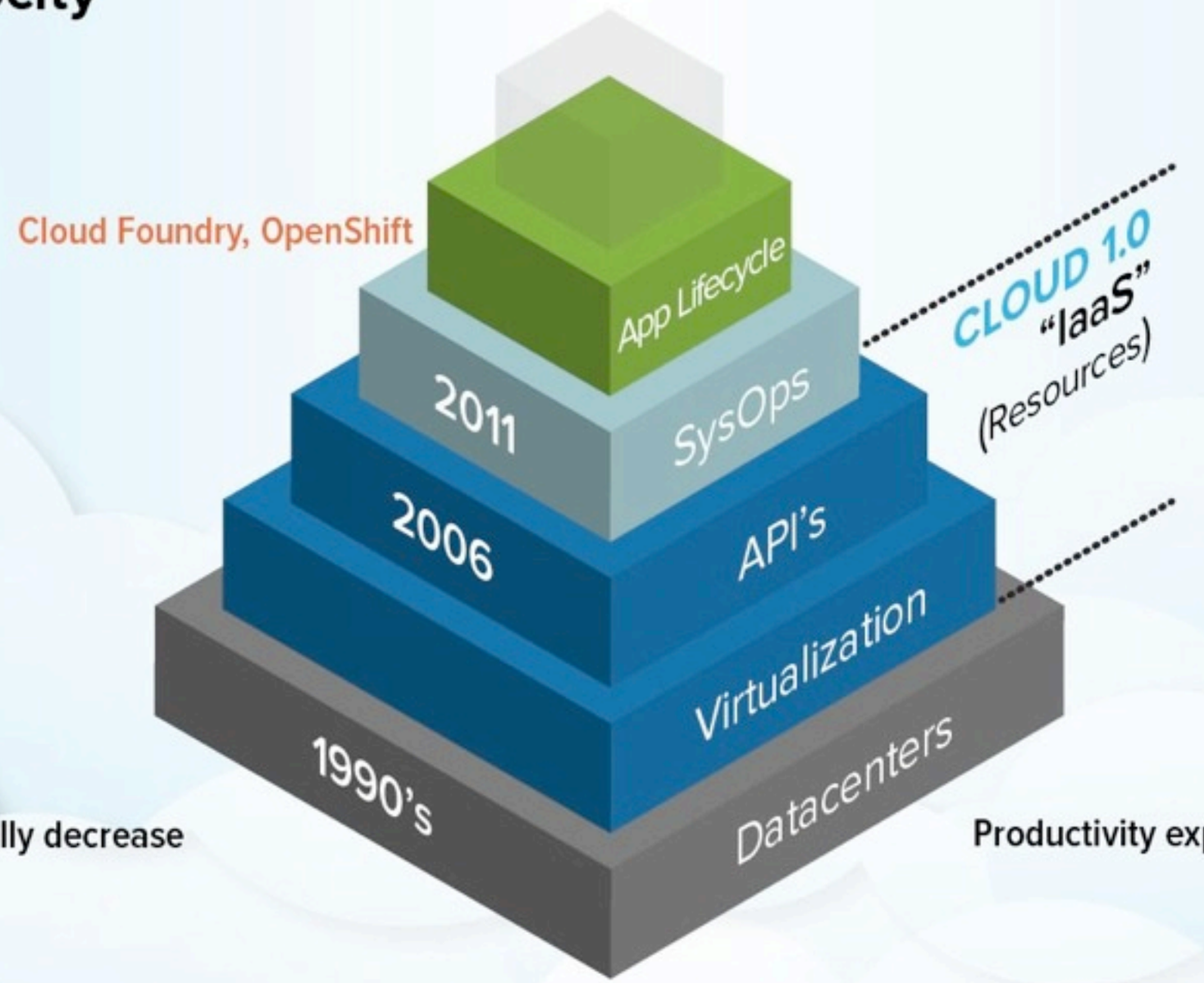


Sources: National Venture Capital Association and Center for Venture Research

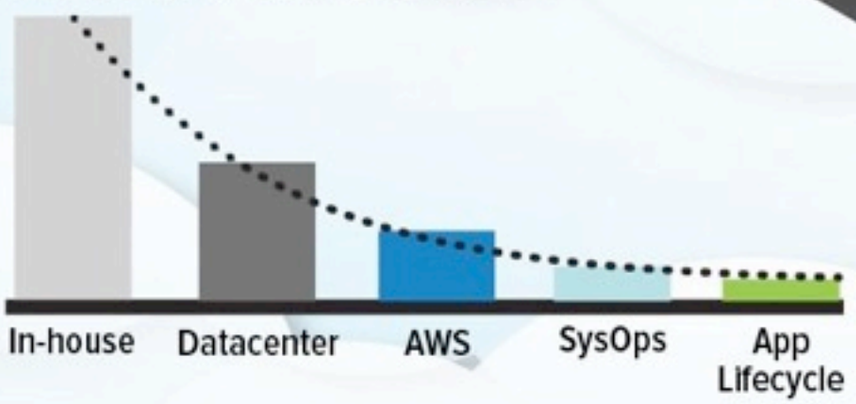
Productivity exponentially increases

Sources: National Venture Capital Association and Center for Venture Research

2012: Velocity



Costs exponentially decrease



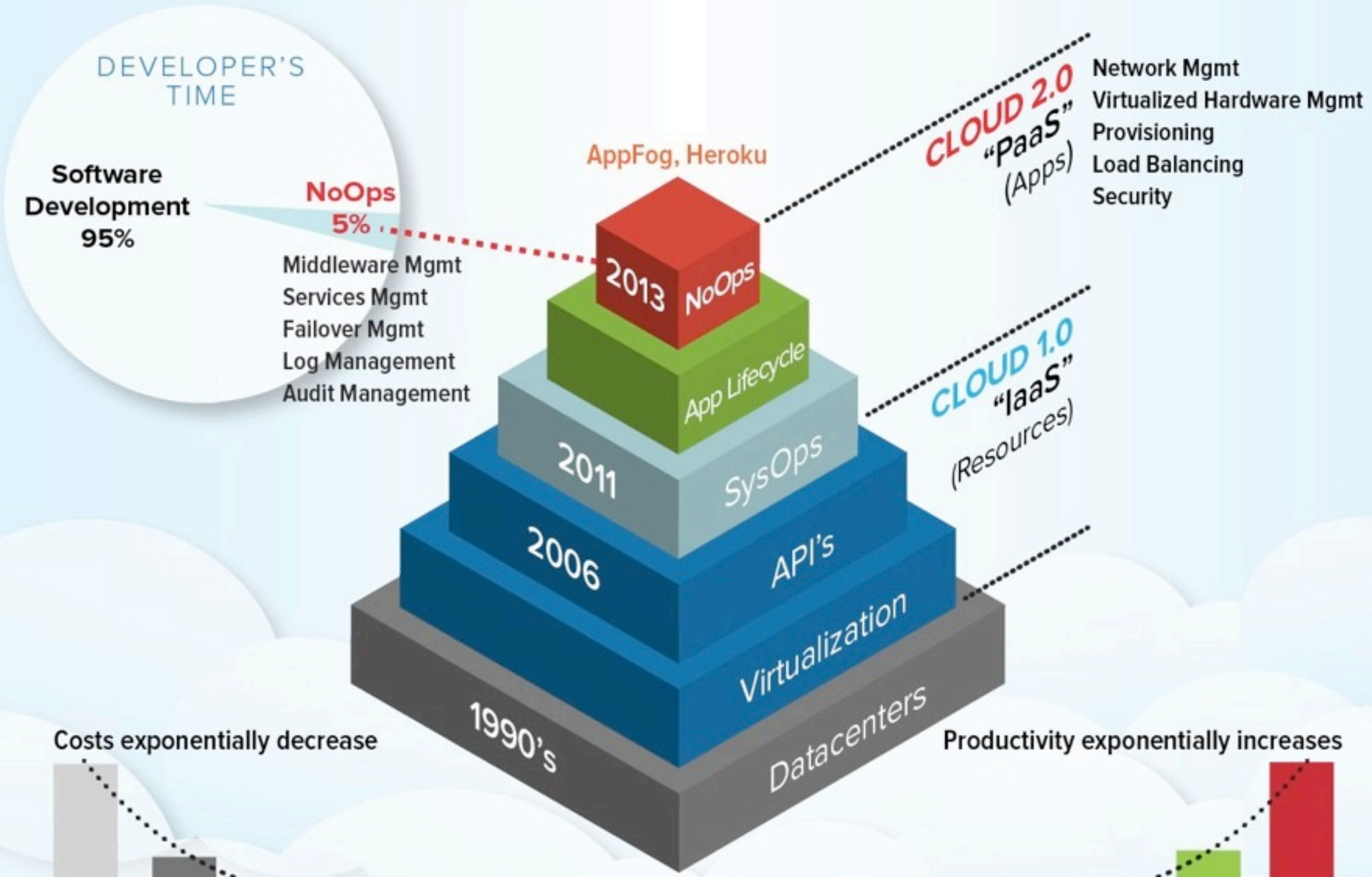
Sources: National Venture Capital Association and Center for Venture Research

Productivity exponentially increases

Sources: National Venture Capital Association and Center for Venture Research

2013: A bright NoOps future

So where does this all lead? The end-game is NoOps. Where building and running an app is purely a developer process — and where developers are not having to spend time doing Ops work.



Costs exponentially decrease

Sources: National Venture Capital Association and Center for Venture Research

Productivity exponentially increases

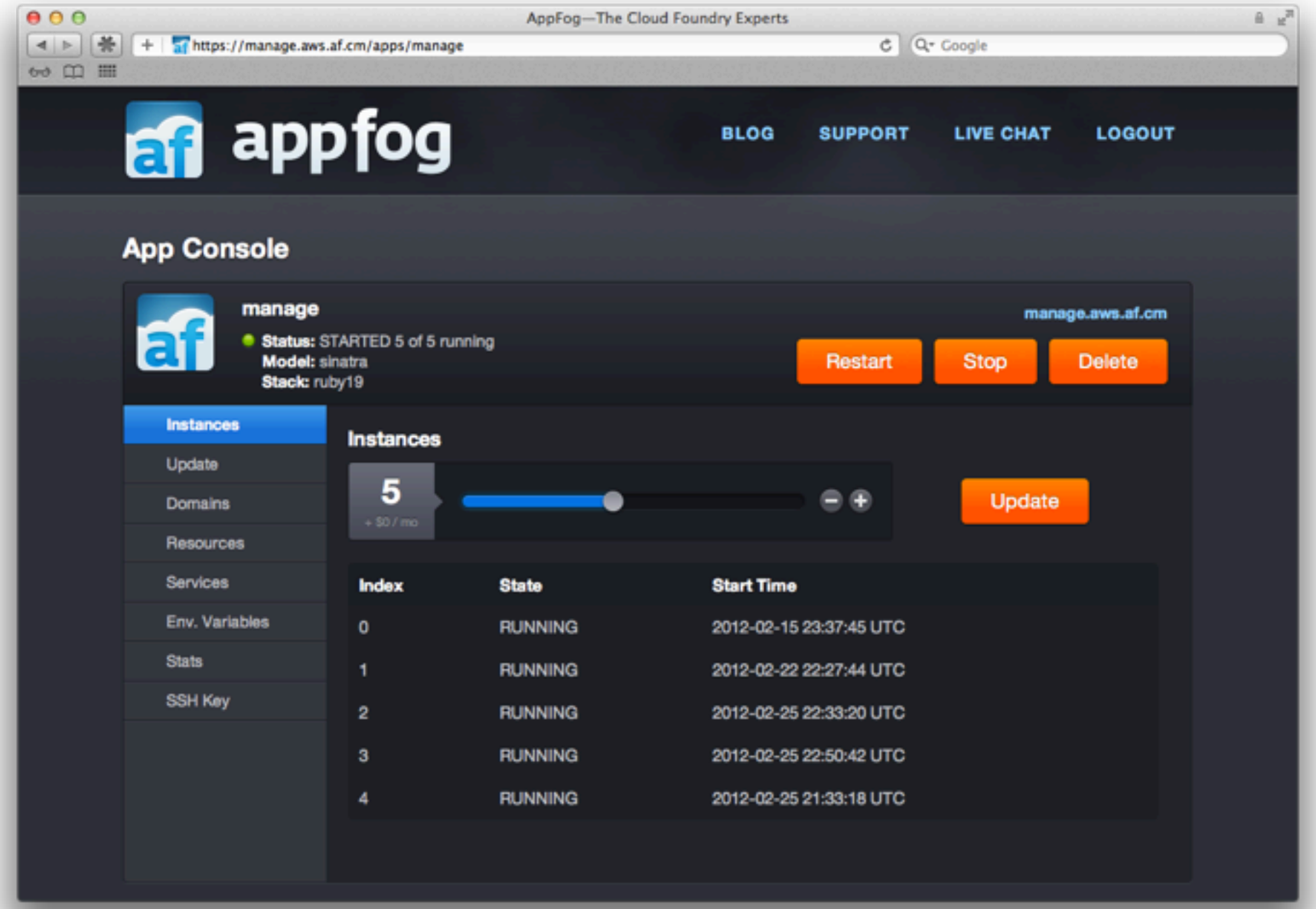
Sources: National Venture Capital Association and Center for Venture Research

Layers of PaaS

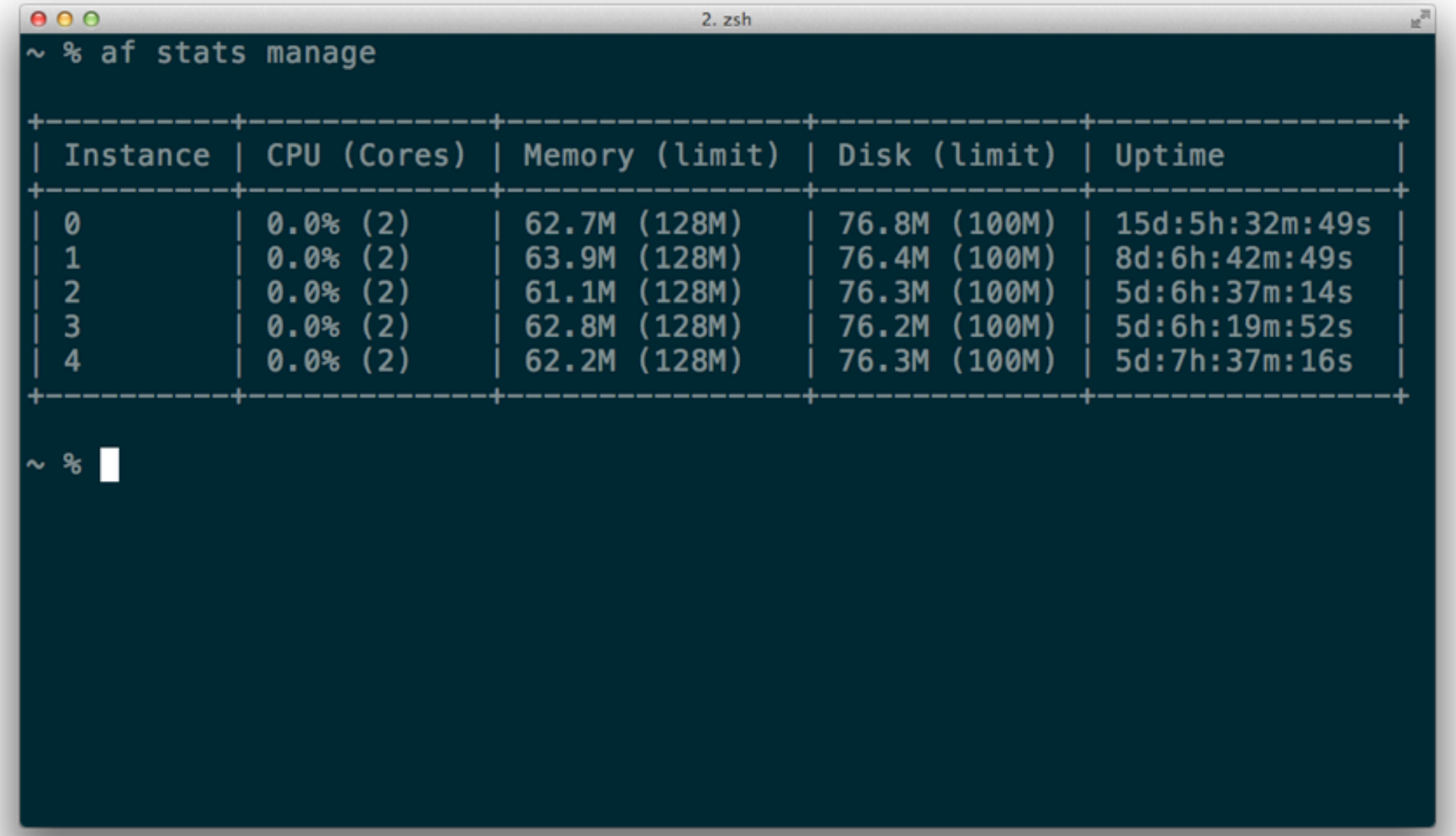
- User experience
- Application lifecycle management
- Orchestration

UX = Developer Experience

- Understand their problems
- Concentrate their efforts
- Help them ship faster!



UX ∋ Interface


2. zsh
~ % af stats manage
+----------+-------------+-----------------+--------------+-----------------+
| Instance | CPU (Cores) | Memory (limit) | Disk (limit) | Uptime |
+----------+-------------+-----------------+--------------+-----------------+
0	0.0% (2)	62.7M (128M)	76.8M (100M)	15d:5h:32m:49s
1	0.0% (2)	63.9M (128M)	76.4M (100M)	8d:6h:42m:49s
2	0.0% (2)	61.1M (128M)	76.3M (100M)	5d:6h:37m:14s
3	0.0% (2)	62.8M (128M)	76.2M (100M)	5d:6h:19m:52s
4	0.0% (2)	62.2M (128M)	76.3M (100M)	5d:7h:37m:16s
+----------+-------------+-----------------+--------------+-----------------+
~ %

UX ∋ Tools

Solve Your Own Problems!

- API for app lifecycle management
- Roll your own tools
- IDE integration
- Continuous Integration
- Autoscaling

UX ∋ Plans and Pricing

- Free plan for dev / test
- What is the fundamental unit?
- No calculators from hell!

UX ∋ Support

- Fast response times
- Comprehensive docs
- Example code
- Community!

Application Lifecycle

- HTTP pipeline
- Language runtimes and libraries
- Services
- App configuration
- Lifecycle events
- Visibility

Orchestration

- Provisioning infrastructure
- Capacity planning / scaling
- Monitoring
- Configuration management

What is CloudFoundry?

CloudFoundry.com
- vs -
CloudFoundry OSS

- CloudFoundry.com is operated by VMware
  - Runs on vSphere
  - In public beta
- CloudFoundry OSS created by VMware
  - http://github.com/cloudfoundry
  - Powers CloudFoundry.com, AppFog and others
- This talk is about CloudFoundry OSS

CloudFoundry is
a kernel for
Application Lifecycle
Management

CloudFoundry Tenets

- Loosely coupled
- Fails fast
- Minimizes single points of failure
- Infrastructure agnostic

Loosely Coupled

- Collection of single-purpose daemons
- Connected by pub/sub
- Distributed state
- Controlled by an HTTP API

Fails Fast

- Optimized for mean-time-to-recovery
- Not mean-time-to-failure

Minimizes Single Points of Failure

- Each component scales horizontally
- (Except for CCDB, HM, NATS)

Infrastructure Agnostic

- Run on your workstation for dev/test
- Run in your data center
- Public Cloud: AWS, Rackspace, HP, Joyent

CF Components

- NATS
- Router
- CloudController
- DEA
- Health Manager
- Service Architecture

NATS

- The “Nervous System”
- Pub/Sub message bus
- Topic + Payload (JSON)
- Subscribe to patterns of topics

Router

- Proxies web requests to backends
- Balances load
- Discovers routes and backends via NATS
  - Eventually consistent
- Proxies apps and CF components alike
- Scales horizontally

CloudController

- REST + JSON API
- Stages and deploys apps
- Single point of truth (CCDB)
  - Users
  - Apps
  - Services

DEA

- "Droplet Execution Agent"
- Starts and stops apps
- Monitors apps
- Announces transitions via NATS

Health Manager

- Reads CloudController's database
- Runloop checks for drift
  - i.e. # instances / app
- Signals to CloudController via NATS

Services Architecture

- Nodes control service installations
  - i.e. MySQL, MongoDB, Postgres, Redis
- Service gateways
  - Expose cluster of nodes to system
  - Gateway abstracts CF-hosted services

Scenario: deployments

- Client POSTs app metadata
- Client sends resource manifest, missing files, binds services
- CC stages app
- CC requests DEA with capacity / capabilities
- First DEA to respond wins, pulls app package
- DEA starts app, signals NATS
- Router discovers app, proxies HTTP

Binding to Services

```
$ head wp-config.php
<?php
// ** Consume service configuration ** //
$services = getenv("VCAP_SERVICES");
$services_json = json_decode($services,true);
$mysql_config = $services_json["mysql-5.1"][0]["credentials"];

// ** MySQL settings from resource descriptor ** //
define('DB_NAME', $mysql_config["name"]);
define('DB_USER', $mysql_config["user"]);
define('DB_PASSWORD', $mysql_config["password"]);
define('DB_HOST', $mysql_config["hostname"]);
define('DB_PORT', $mysql_config["port"]);
// ** MySQL settings from resource descriptor ** //

$ af bind-service my-service my-app
```

Binding to Ports

```
var app = express.createServer();
// app definition elided
var port = process.env.VCAP_APP_PORT || 8001;
app.listen(port);
```

CloudFoundry at AppFog

phpfog Lessons Learned

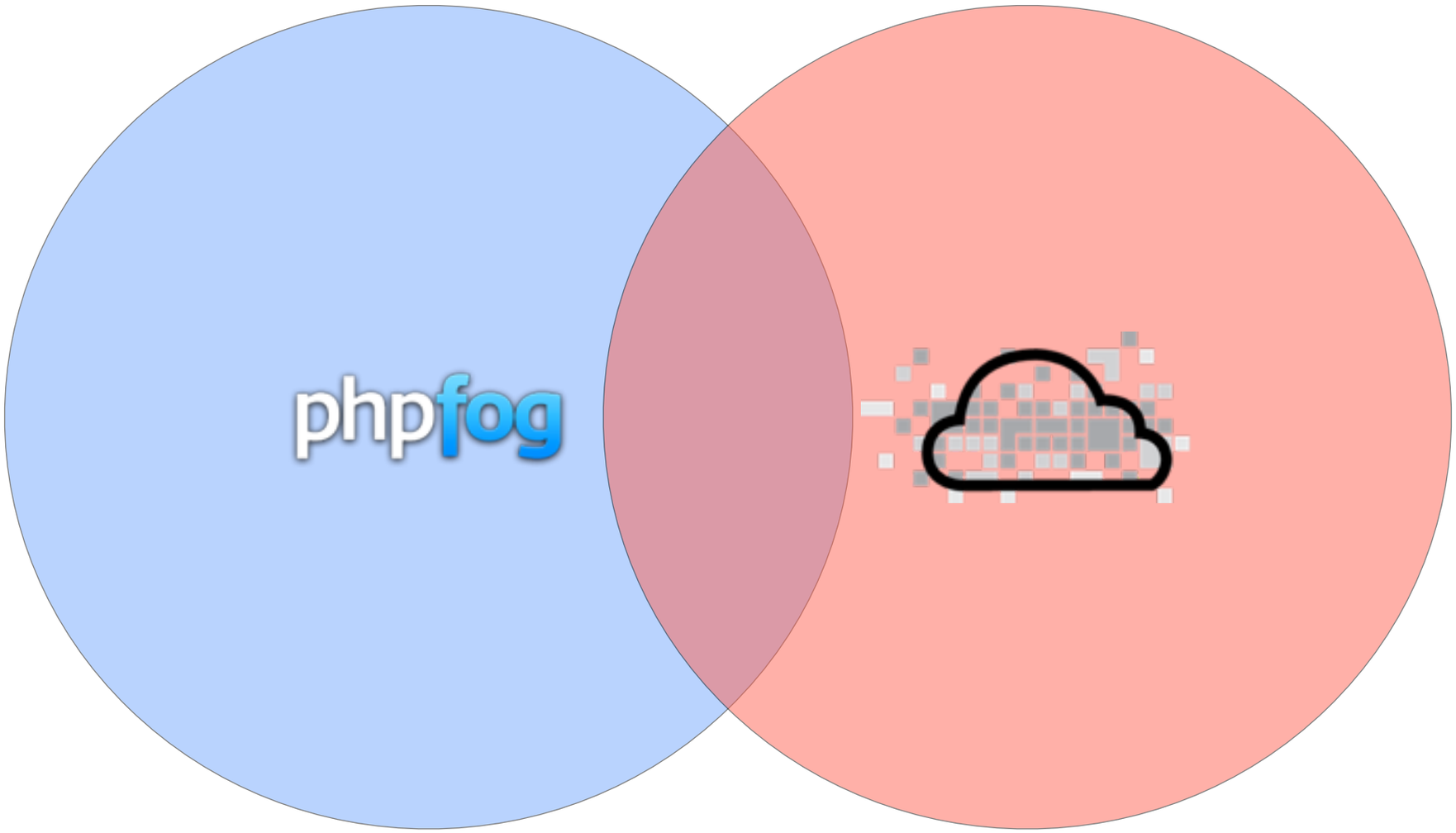
- AppFog is our second PaaS product
- We built our first PaaS, PHP Fog, from first principles
- We built all three layers from scratch
- We learned a lot!

Up the ante: create the best multi-language, multi-service, multi-infrastructure platform

Why did we choose CloudFoundry?

- Embraces polyglot programming, polyglot persistence
- Focus our energies on UX, orchestration
- Excellent code quality
- Vibrant OSS community

PHP Fog / CloudFoundry Architecture Comparison



Similarities

- Provides N-tier architecture for PHP apps
- Isolates apps for multi-tenant
- Scales horizontally
- Manages app configuration

PHP Fog UX

- Comprehensive management console
- New API!
- Tiered pricing model
  - Fundamental unit is the dedicated VM
  - Easy plan changes, a la carte upgrades

PHP Fog App Lifecycle

- PHP-specific
- Every app gets a MySQL database
    - (More services available as add-ons)
- One URL per app
- Git deployments

PHP Fog Lifecycle Innovations

- HTTP caching tier (Varnish)
- Wildcard subdomains
- SSL termination for custom domains
- Dedicated app servers

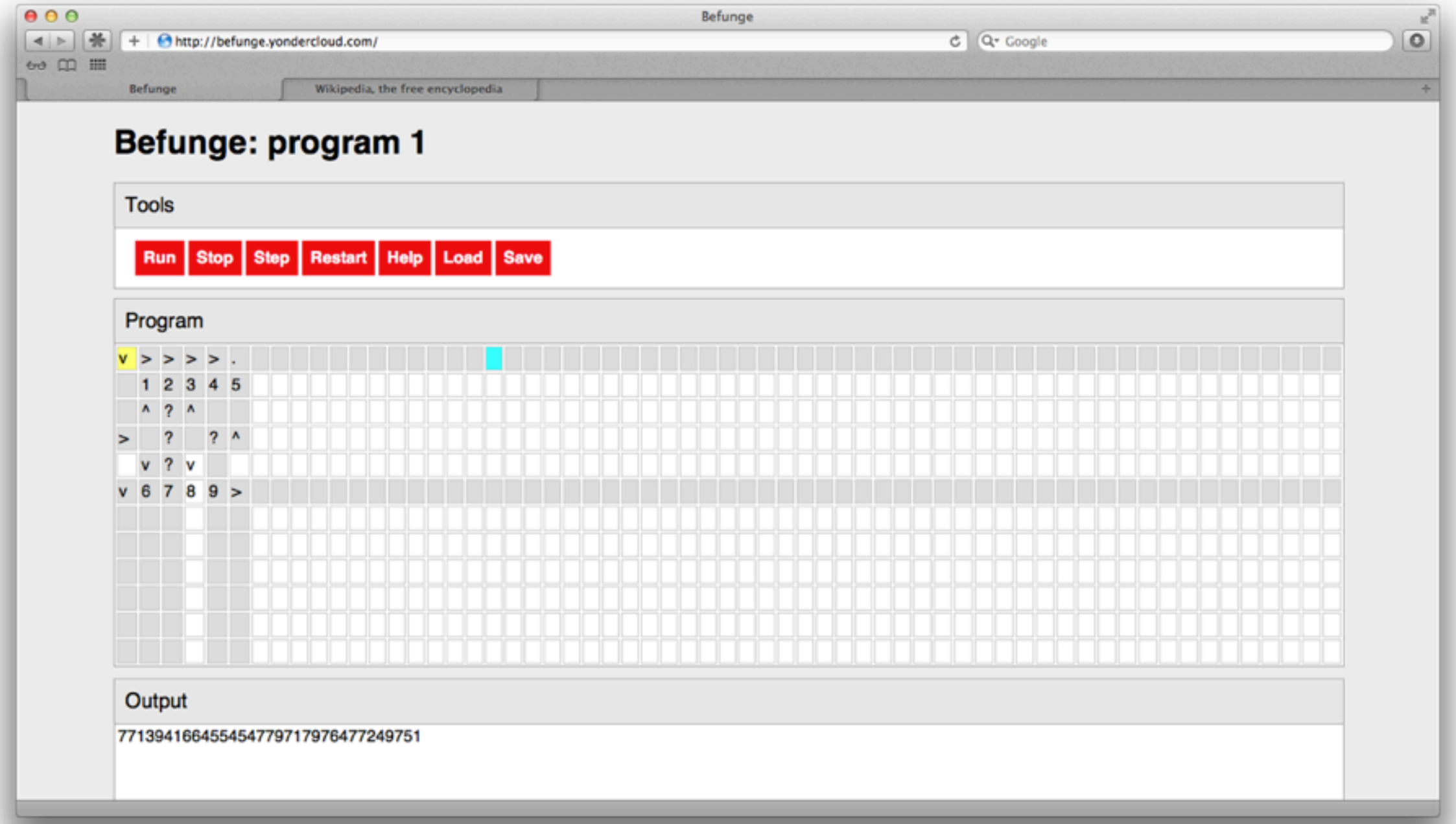
PHP Fog Orchestration

- Config management (Puppet)
- Ad hoc scripting
- Web UI

Where are we now?

- Next generation of our web console
- HTTP caching tier for CF
- PHP / Apache2 runtime support
- Orchestration for multiple public clouds
- Lots of example apps and spinoffs!

(spinoff)



https://github.com/lhitchon/befunge

Supported Technologies

- Runtimes: PHP, Ruby (1.8, 1.9), NodeJS
- Services: MySQL, MongoDB

What's Next?

- AppFog exits private beta, opens registration
- Support for PHP Fog's add-on ecosystem
- More infrastructure choices
- More services and runtimes

8 accepted pull requests (and counting!)

http://octodex.github.com/constructocat-v2

Thank you!

Jeremy Voorhis
Senior Engineer, AppFog Inc
jeremy@appfog.com
@jvoorhis
http://www.appfog.com