Introduction

Hewlett Packard Enterprise

Aaron Lefkowitz
Engineering Manager
Helion Cloud Foundry

Vlad Iovanov
Technical Lead
Helion Cloud Foundry
Why?
Can’t we just use BOSH?

• BOSH is about Virtual Machines

• We wanted something for containers

• For Science!
Separate all the things

• BOSH and CF are tightly coupled
• BOSH technology choices limit implementations
• ERB and Monit are hard to dislodge
What’s in a container?

- Start FROM an Ubuntu 14.04 Docker image
- Create a “stemcell”-like layer on top of it
- Add packages, jobs
- Write an ENTRYPOINT and add some tooling for configuration
Build the World!

- It happens at build time
- It happens in parallel
- It builds in a container that has compilation dependencies
  - BOSH has both runtime and compilation dependencies
- It does smart detection of dependencies
  - Only builds what is required by the role manifest
The Assembly Line

- Role Manifest
- Opinions
- BOSH Releases

- Role Manifest
- Opinions
- BOSH release #1
- BOSH release #2
- BOSH release #N

fissile

Docker Images
Configuration – the fine print

Role Manifest

---
roles:
- name: nats
  jobs:
  - name: nats
    release_name: cf
  - name: nats_stream_forwarder
    release_name: cf
  - name: metron_agent
    release_name: cf
configuration:
  templates:
    index: "0"
  networks.default.ip: "((IP_ADDRESS))"
  networks.default.dns_record_name: "((DNS_RECORD_NAME))"
  properties.nats.user: "((NATS_USER))"
  properties.nats.password: "((NATS_PASSWORD))"

Opinions

---
properties:
nats:
  debug: false
  monitor_port: 0
  port: 4222
  prof_port: 0
  trace: false

“Dark” Opinions

---
properties:
nats:
  password: ""
Run, Docker, Run!

run.sh

- Execute hook scripts
  - Run configgin to process templates
    - Start rsyslog and cron
      - Start monit
        - Trap INT and TERM
Configuration

- Configuring Cloud Foundry is hard
- Wanted to distill this into something smaller for the user
- Configgin augments bosh_template
- Mustache templates eliminate configuration complexity

<table>
<thead>
<tr>
<th>Name</th>
<th>#</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>specs</td>
<td>500</td>
<td>Each job has spec files</td>
</tr>
<tr>
<td>opinions</td>
<td>200</td>
<td>Our curated defaults which started from BOSH manifest generation</td>
</tr>
<tr>
<td>user global</td>
<td>90</td>
<td>Values set by a user specific to their environment</td>
</tr>
<tr>
<td>user role</td>
<td>20</td>
<td>Values set by a user specific to their environment AND a specific VM</td>
</tr>
</tbody>
</table>
Configuration

Layered dynamic

- Consul keyspace for each of: spec, opinions, job, role

Problems:

- Slow to run
- Still have to restart, so dynamic doesn’t matter
- Required yet another KV Raft process

Layered static

- Environment variables for user values
- Everything else pre-computed and stored in each container
Pull Requests

- DNS Lookup changes
- Hardcoded values that are subtle BOSHisms
- Touching `/proc` without restraint

github.com/cloudfoundry/vcap-common/pull/19
github.com/cloudfoundry/warden/pull/103
github.com/cloudfoundry/cf-release/pull/915
github.com/cloudfoundry/cf-release/pull/928
github.com/cloudfoundry-incubator/etcd-release/pull/9
github.com/cloudfoundry-incubator/consul-release/pull/17
github.com/cloudfoundry-incubator/garden-linux-release/pull/19
github.com/cloudfoundry-incubator/garden-runc-release/pull/7
github.com/cloudfoundry-incubator/diego-release/pull/146

... and more
Demo
From zero to deployed in a few minutes
Demo
Reaping the benefits
The end
Wrapping up…

– We’re open sourcing fissile and configgin (Apache 2.0)
– We want to:
  – Add support for other base images
  – We want to improve layering
  – Remove monit as a sticking point
  – We want to continue the effort of decoupling CF from BOSH
– Please contribute - we can make it better, faster

What are we releasing?
- The tooling, no images