What is BOSH?

BOSH is like chef

...if it were a completely different product

(it actually manages VMs)

It's like Amazon CloudFormation

...if it supported more than AWS

It's like Canonical Juju

...if it supported more than Ubuntu

Guilty!

"We" wrote our own deployment tool…

But why???

Declarative

Idempotent

Scalable to huge clusters

Cloud agnostic

O.S. agnostic

What has been deployed with BOSH

Cloud Foundry

Pivotal hadoop

RabbitMQ cluster

Mysql

Redis

Riak cluster

Cassandra

Postgres

Jenkins

ETCd

MariaDB

SkyDNS

Zookeeper

...

Words

- Release
- Stemcell
- Deployment

A release is code

- Packages
  - Include src code
  - Includes blobs
  - Depend on other packages
- Jobs
  - Use packages
  - start script
  - monit

```
± ds |master ✗| → bosh releases

+-------------+-----------+---------------+
| Name        | Versions  | Commit Hash   |
+-------------+-----------+---------------+
| cf          | 156*      | d406f808+     |
| cf-mysql    | 6*        | 39f97623+     |
| cf-redis    | 39*       | 2f70a822+     |
| riak-cs-dev | 0.15-dev* | 8220a50f+     |
+-------------+-----------+---------------+
(*) Currently deployed
(+) Uncommitted changes

Releases total: 4
```

- blobs
  - erlang
  - git
  - libyaml
  - riak
  - riak-cs
  - ruby
  - stanchion
- config
- dev_releases
- jobs
  - cf-riak-cs-broker
  - riak-cs
  - stanchion
  - .gitkeep
- packages
  - cf-riak-cs-broker
  - erlang
  - git
  - riak
  - riak-cs
  - stanchion
  - .gitkeep
  - libyaml
  - ruby
- src
  - cf-riak-cs-broker
  - .gitkeep

A stemcell is a machine image

That can become anything… so it's basically just the OS + an agent

```
± ds |master x| → bosh stemcells

+----------------------------+---------+-----------------------------------------------+
| Name                       | Version | CID                                           |
+----------------------------+---------+-----------------------------------------------+
| bosh-stemcell              | 993     | stemcell-b6072772-cae8-468c-9779-5c8df30d02d9 |
| bosh-warden-boshlite-ubuntu | 24*    | stemcell-49f42459-dbe7-4464-9f08-51c17be24726 |
+----------------------------+---------+-----------------------------------------------+

(*) Currently in-use

Stemcells total: 2
```

Deployment = cluster

"Jobs", properties, networks, resource pools

```
± ds |master ✗| → bosh deployments

+----------------+-------------------+-------------------------------+
| Name           | Release(s)        | Stemcell(s)                   |
+----------------+-------------------+-------------------------------+
| cf-redis       | cf-redis/39       | bosh-warden-boshlite-ubuntu/  |
+----------------+-------------------+-------------------------------+
| cf-warden      | cf/156            | bosh-warden-boshlite-ubuntu/  |
+----------------+-------------------+-------------------------------+
| cf-warden-mysql | cf-mysql/6       | bosh-warden-boshlite-ubuntu/  |
+----------------+-------------------+-------------------------------+
| riak-cs-service | riak-cs-dev/0.15-dev | bosh-warden-boshlite-ubuntu/ |
+----------------+-------------------+-------------------------------+

Deployments total: 4
```
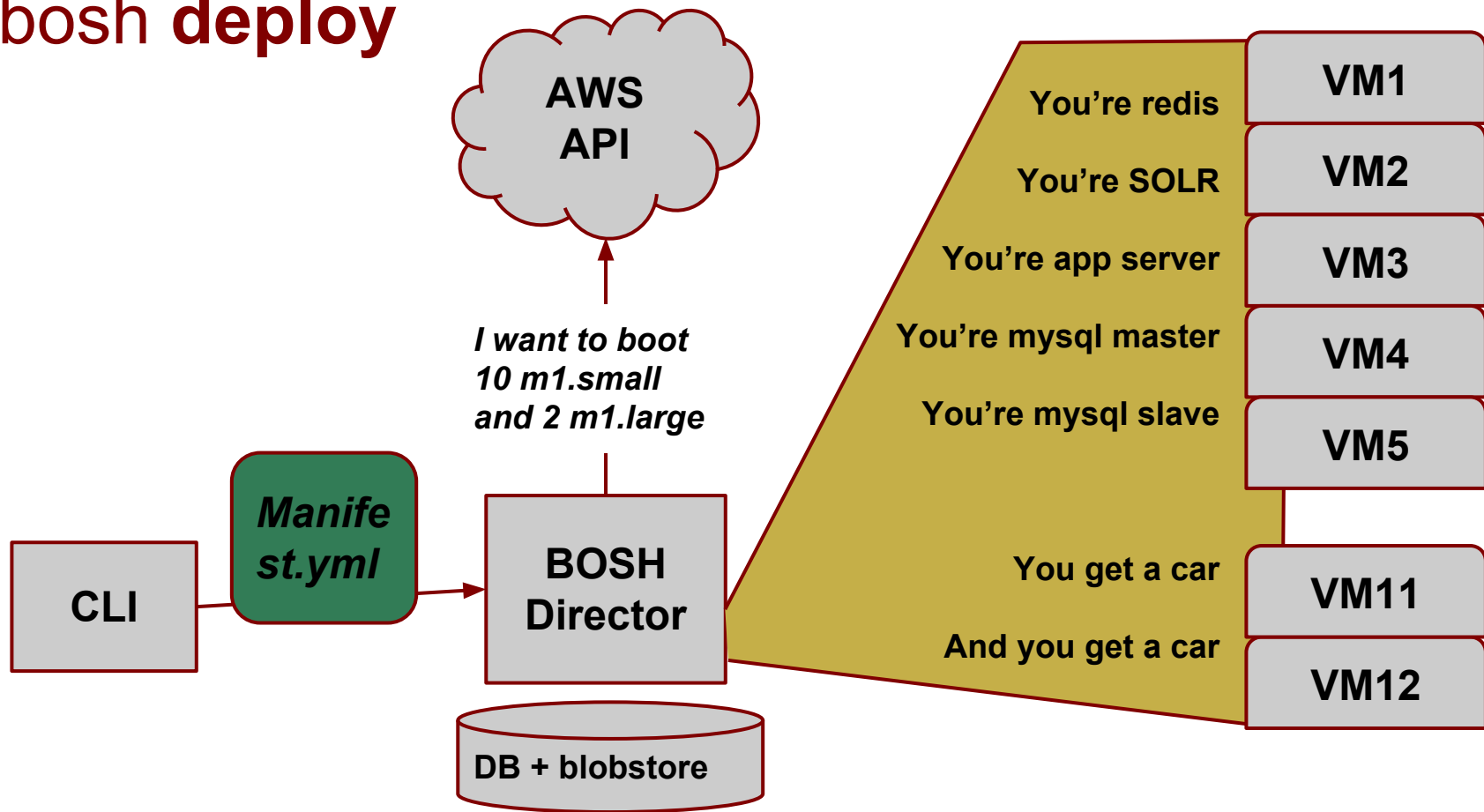
```yaml
jobs:
- instances: 1
  name: mysql
  networks:
  persistent_disk: 100000
  properties:
  release: cf-mysql
  resource_pool: services-small
  template:
- instances: 1
  name: cf-mysql-broker
  networks:
  properties:
  release: cf-mysql
  resource_pool: services-small
  template:
name: cf-warden-mysql
networks:
properties:}
releases:
resource_pools:
- cloud_properties:
    name: random
  name: services-small
  network: services1
  size: 2
  stemcell:
    name: bosh-warden-boshlite-ubuntu
    version: latest
update:
  canaries: 1
  canary_watch_time: 30000-180000
  max_in_flight: 4
  update_watch_time: 30000-180000
```

BOSH compiles everything

As part of deployment…

Your release code is compiled for the stemcell

bosh **deploy**

AWS
API

*I want to boot
10 m1.small
and 2 m1.large*

CLI

*Manife
st.yml*

**BOSH
Director**

DB + blobstore

You're redis — **VM1**

You're SOLR — **VM2**

You're app server — **VM3**

You're mysql master — **VM4**

You're mysql slave — **VM5**

You get a car — **VM11**

And you get a car — **VM12**

bosh **deploy**

Bind templates

Compile packages

Start new VMs

Update jobs

Delete unused VMs

```
Compiling packages
  common/1 (00:00:06)
  mysqlclient/2 (00:01:57)
  ruby/4 (00:04:59)
  cf-mysql-broker/6 (00:00:27)
  mysql/1 (00:20:18)
Done                    5/5 00:20:18

Preparing DNS
  binding DNS (00:00:00)
Done                    1/1 00:00:00

Creating bound missing VMs
  services-small/0 (00:00:04)
  services-small/1 (00:00:04)
Done                    2/2 00:00:04

Binding instance VMs
  cf-mysql-broker/0 (00:00:01)
  mysql/0 (00:00:01)
Done                    2/2 00:00:01

Preparing configuration
  binding configuration (00:00:00)
Done                    1/1 00:00:00

  Started updating job mysql: mysql/0 (canary)
     Done updating job mysql: mysql/0 (canary)
  Started updating job cf-mysql-broker: cf-mysql-broker/0 (canary)
     Done updating job cf-mysql-broker: cf-mysql-broker/0 (canary)

Task 54 done
Started         2014-02-19 04:02:25 UTC
Finished        2014-02-19 04:22:50 UTC
Duration        00:20:25
```

bosh **deploy**

Simple case

Only update and restart jobs that have changed (or require changed properties)

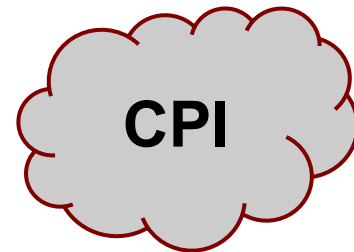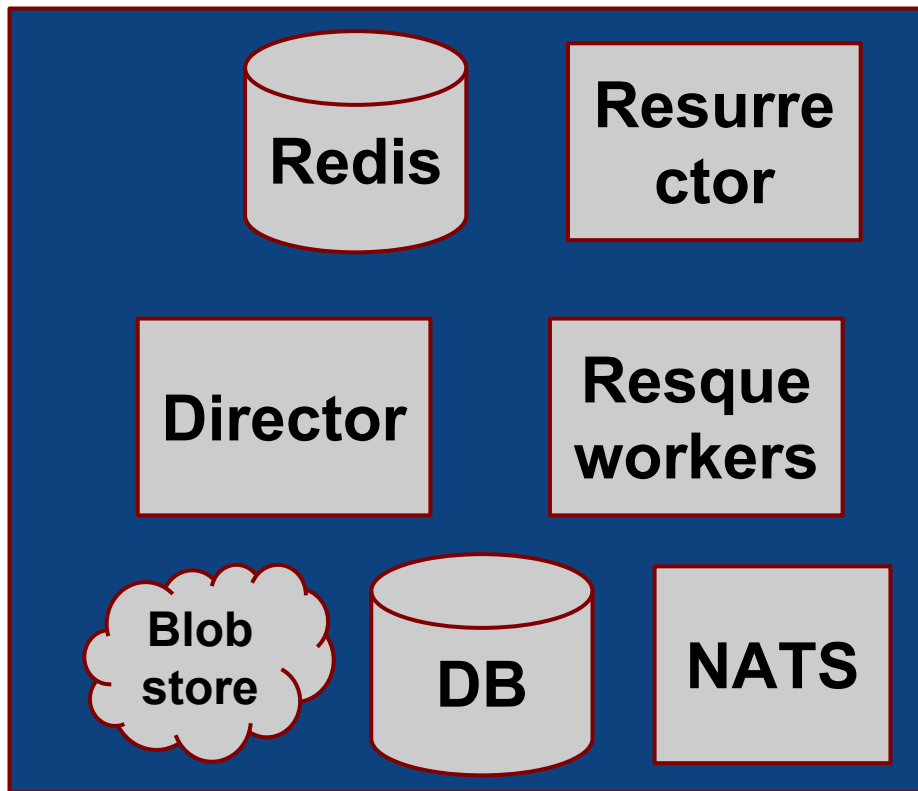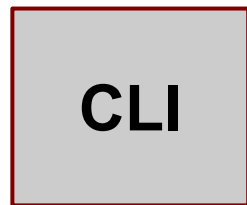Canaries & MaxIF

Stemcell upgrade

Detach persistent disk

Recreate VM
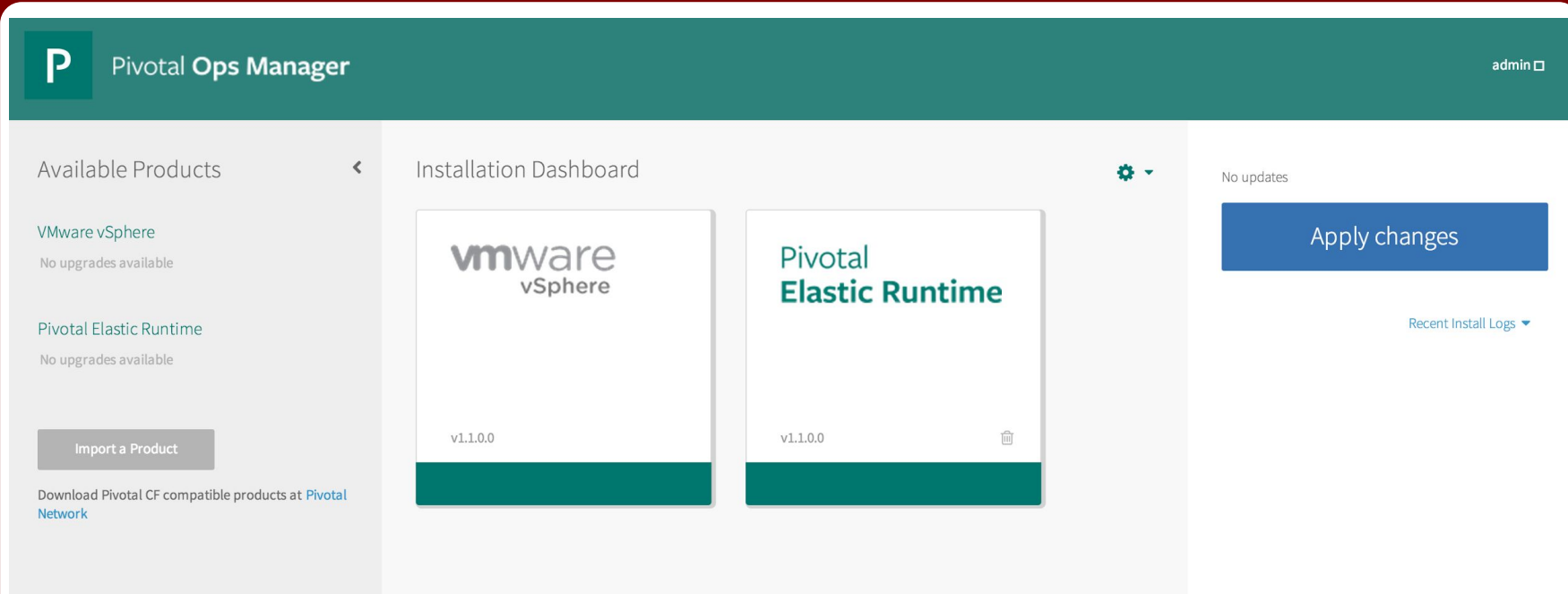
Attach disk

Update jobs

Start jobs

Components of BOSH

So BOSH is a distributed system itself.

How do you think we deploy BOSH?

We made a GUI in front of BOSH

cause it's hard to learn bosh

BOSH-lite… now with fewer calories

Instead of creating a VM, we create a linux container based on cgroups

https://github.com/cloudfoundry/bosh-lite

You can finally try deploying something with bosh on your mac using vagrant