Before we get started...
Install Prerequisites

- VirtualBox (https://www.virtualbox.org/wiki/Downloads)
- Vagrant (http://www.vagrantup.com/downloads.html)
- Ruby 1.9.3 - use one of these version managers:
  - RVM (http://rvm.io)
  - rbenv (https://github.com/sstephenson/rbenv)
- BOSH CLI Gem (gem install bosh_cli)
- Git (http://git-scm.com/downloads) or “brew install git”
- spiff (https://github.com/cloudfoundry-incubator/spiff/releases)
- cf (https://github.com/cloudfoundry/cli/releases)
After all of that:


If you want to see what this is doing first:

Architecture and Operations

- Install Prerequisites
- Install Cloud Foundry Runtime on BOSH Lite
- Cloud Foundry Architecture Overview
- Cloud Foundry BOSH Overview
- Test Your New Private Cloud Foundry!
Cloud Foundry Architecture Stack

- Router
- Cloud Controller
- UAA/Login Servers
- Health Manager
- Service Broker Node(s)
- DEA Pool
- Loggregator
- Messaging (NATS)
- BOSH Director
- BOSH Agent

IaaS
Deploying Applications to Cloud Foundry Runtime

1. Upload bits/metadata
2. Create/bind services
3. Stage app
4. Deploy app
Stage an App

Cloud Controller

DEA

System Buildpacks 
Detect Yes

Compile +
Upload =

Runtime
Deploy and Scale an App

Blobstore → Cloud Controller

Access App → Router → Messaging (NATS) → DEA → DEA → DEA

Runtime
Create and Bind Services

Create service (HTTP)
bind service (HTTP)
reserve resources
obtain connection data
Maintain App Health

- Blobstore
- Cloud Controller
- Messaging (NATS)
- Health Manager
- DEA
- DEA
- DEA
- Desired State
- Actual State
- Runtime
Monitor App Logs
Architecture and Operations

- Install Prerequisites
- Install Cloud Foundry Runtime on BOSH Lite
- Cloud Foundry Architecture Overview
- Cloud Foundry BOSH Overview
- Test Your New Private Cloud Foundry!
Cloud Foundry Architecture: Why?

- Component Isolation
- Scalability
- Fault Tolerance
- Pre-provisioned Capacity (Containers/Warden)
How do we manage this thing?
We need a toolchain that can manage a large distributed system through:

- Deployment
- Configuration Changes
- Updates/Upgrades (w/ minimal - zero - downtime!)
- Component Failure / Restoration of Service
- Scale Out / Scale In
- Across multiple IaaS providers: vSphere, OpenStack, AWS, Apache Cloudstack, Google Compute Engine, and beyond...
Cloud Foundry BOSH
Why BOSH?

- Provision services, not machines
- Eliminate bespoke automation on top of configuration management
- Enable continuous delivery of platform services
- Cloud-agnostic view of platform operations
- Holistic toolchain to “rule them all”
- How we manage Cloud Foundry in production!
<table>
<thead>
<tr>
<th>Component</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router</td>
<td></td>
</tr>
<tr>
<td>Cloud Controller</td>
<td></td>
</tr>
<tr>
<td>UAA/Login Servers</td>
<td></td>
</tr>
<tr>
<td>Health Manager</td>
<td></td>
</tr>
<tr>
<td>Service Broker Node(s)</td>
<td></td>
</tr>
<tr>
<td>DEA Pool</td>
<td></td>
</tr>
<tr>
<td>Loggregator</td>
<td></td>
</tr>
<tr>
<td>Messaging (NATS)</td>
<td></td>
</tr>
<tr>
<td>BOSH Director</td>
<td></td>
</tr>
<tr>
<td>BOSH Agent</td>
<td></td>
</tr>
<tr>
<td>IaaS</td>
<td></td>
</tr>
</tbody>
</table>
**BOSH in Action**

- **Deploy my Services**
  - DB
  - Blobstore
  - Health Monitor

- **BOSH Director**

- **NATS**

- **Worker VMs**
  - Messaging
  - Health Manager
  - Cloud Controller
  - Target VM

- **IaaS**
BOSH: Cloud Provider Interface

**Stemcell**
- `create_stemcell(image, cloud_properties)`
- `delete_stemcell(stemcell_id)`

**VM**
- `create_vm(agent_id, stemcell_id, resource_pool, networks, disk_locality, env)`
- `delete_vm(vm_id)`
- `reboot_vm(vm_id)`
- `configure_networks(vm_id, networks)`

**Disk**
- `create_disk(size, vm_locality)`
- `delete_disk(disk_id)`
- `attach_disk(vm_id, disk_id)`
- `detach_disk(vm_id, disk_id)`
BOSH Lite

https://github.com/cloudfoundry/bosh-lite
Architectures and Operations

- Install Prerequisites
- Install Cloud Foundry Runtime on BOSH Lite
- Cloud Foundry Architecture Overview
- Cloud Foundry BOSH Overview
- Test Your New Private Cloud Foundry!
Completed BOSH Lite Deploy

Done updating job ha_proxy_z1 > ha_proxy_z1/0 (00:00:28)
Done updating job nats_z1 > nats_z1/0 (00:00:31)
Done updating job etcd_z1 > etcd_z1/0 (00:00:32)
Done updating job postgres_z1 > postgres_z1/0 (00:00:32)
Done updating job router_z1 > router_z1/0 (00:00:36)
Done updating job uaa_z1 > uaa_z1/0 (00:00:37)
Done updating job login_z1 > login_z1/0 (00:00:39)
Done updating job loggregator_trafficcontroller_z1 > loggregator_trafficcontroller_z1/0 (00:01:08)
 Done updating job loggregator_z1 > loggregator_z1/0 (00:01:08)
 Done updating job api_z1 > api_z1/0 (00:01:16)
 Done updating job hm9000_z1 > hm9000_z1/0 (00:01:27)
 Done updating job runner_z1 > runner_z1/0 (00:02:26)

Task 3 done

Started 2014-06-25 04:33:47 UTC
Finished 2014-06-25 04:44:04 UTC
Duration 00:10:17

Deployed `cf-manifest.yml' to `Bosh Lite Director'
BOSH Status

Config
    /Users/pivotal/.bosh_config

Director
    Name       Bosh Lite Director
    URL        https://192.168.50.4:25555
    Version    1.2559.0 (fe0b2436)
    User       admin
    UUID       c5f8d0da-f8ac-4918-a1a3-0a846fb97d09
    CPI        warden
    dns        disabled
    compiled_package_cache enabled (provider: local)
    snapshots  disabled

Deployment
    Manifest   /Users/pivotal/bosh-lite-tutorial/bosh-lite/manifests/cf-manifest.yml
    Deployment `cf-warden'
"bosh vms --details"

Director task 4

Task 4 done

<table>
<thead>
<tr>
<th>Job/index</th>
<th>State</th>
<th>Resource Pool</th>
<th>IPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>api_z1/0</td>
<td>running</td>
<td>large_z1</td>
<td>10.244.0.138</td>
</tr>
<tr>
<td>etcd_z1/0</td>
<td>running</td>
<td>medium_z1</td>
<td>10.244.0.42</td>
</tr>
<tr>
<td>ha_proxy_z1/0</td>
<td>running</td>
<td>router_z1</td>
<td>10.244.0.34</td>
</tr>
<tr>
<td>hm9000_z1/0</td>
<td>running</td>
<td>medium_z1</td>
<td>10.244.0.142</td>
</tr>
<tr>
<td>loggregator_trafficcontroller_z1/0</td>
<td>running</td>
<td>small_z1</td>
<td>10.244.0.10</td>
</tr>
<tr>
<td>loggregator_z1/0</td>
<td>running</td>
<td>medium_z1</td>
<td>10.244.0.14</td>
</tr>
<tr>
<td>login_z1/0</td>
<td>running</td>
<td>medium_z1</td>
<td>10.244.0.134</td>
</tr>
<tr>
<td>nats_z1/0</td>
<td>running</td>
<td>medium_z1</td>
<td>10.244.0.6</td>
</tr>
<tr>
<td>postgres_z1/0</td>
<td>running</td>
<td>medium_z1</td>
<td>10.244.0.30</td>
</tr>
<tr>
<td>router_z1/0</td>
<td>running</td>
<td>router_z1</td>
<td>10.244.0.22</td>
</tr>
<tr>
<td>runner_z1/0</td>
<td>running</td>
<td>runner_z1</td>
<td>10.244.0.26</td>
</tr>
<tr>
<td>uaa_z1/0</td>
<td>running</td>
<td>medium_z1</td>
<td>10.244.0.130</td>
</tr>
</tbody>
</table>

VMs total: 12
bosh-lite/scripts/add-route

Adding the following route entry to your local route table to enable direct warden container access. Your sudo password may be required.
- net 10.244.0.0/19 via 192.168.50.4
add net 10.244.0.0: gateway 192.168.50.4
Setting up “me” org, “test” space...

Setting api endpoint to https://api.10.244.0.34.xip.io...
OK

API endpoint: https://api.10.244.0.34.xip.io (API version: 2.2.0)
Not logged in. Use 'cf login' to log in.
API endpoint: https://api.10.244.0.34.xip.io
Authenticating...
OK
Use 'cf target' to view or set your target org and space
Creating org me as admin...
OK

TIP: Use 'cf target -o me' to target new org
API endpoint: https://api.10.244.0.34.xip.io (API version: 2.2.0)
User: admin
Org: me
Space: No space targeted, use 'cf target -s SPACE'
Creating space test in org me as admin...
OK
Assigning role SpaceManager to user admin in org me / space test as admin...
OK
Assigning role SpaceDeveloper to user admin in org me / space test as admin...
OK
It goes on to push an app called: “cf-env”
cf-env successfully pushed...

Showing health and status for app cf-env in org me / space test as admin...
OK

requested state: started
instances: 1/1
usage: 128M x 1 instances
urls: env.10.244.0.34.xip.io

<table>
<thead>
<tr>
<th>state</th>
<th>since</th>
<th>cpu</th>
<th>memory</th>
<th>disk</th>
</tr>
</thead>
<tbody>
<tr>
<td>#0 running</td>
<td>2014-06-24 11:44:59 PM</td>
<td>0.0%</td>
<td>71.6M of 128M</td>
<td>0 of 1G</td>
</tr>
</tbody>
</table>

Getting apps in org me / space test as admin...
OK

<table>
<thead>
<tr>
<th>name</th>
<th>requested state</th>
<th>instances</th>
<th>memory</th>
<th>disk</th>
<th>urls</th>
</tr>
</thead>
<tbody>
<tr>
<td>cf-env</td>
<td>started</td>
<td>1/1</td>
<td>128M</td>
<td>1G</td>
<td>env.10.244.0.34.xip.io</td>
</tr>
</tbody>
</table>
Cloud Foundry Environment

<table>
<thead>
<tr>
<th>Environment Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUNDLE_BIN_PATH</td>
<td>/home/vcap/app/vendor/bundle/ruby/1.9.1/gems/bundler-1.3.2/bin/bundle</td>
</tr>
<tr>
<td>BUNDLE_GEMFILE</td>
<td>/home/vcap/app/Gemfile</td>
</tr>
<tr>
<td>GEM_HOME</td>
<td>/home/vcap/app/vendor/bundle/ruby/1.9.1</td>
</tr>
<tr>
<td>GEM_PATH</td>
<td></td>
</tr>
<tr>
<td>HOME</td>
<td>/home/vcap/app</td>
</tr>
<tr>
<td>LANG</td>
<td>en_US.UTF-8</td>
</tr>
<tr>
<td>MEMORY_LIMIT</td>
<td>128</td>
</tr>
<tr>
<td>OLDPWD</td>
<td>/home/vcap</td>
</tr>
<tr>
<td>PATH</td>
<td>/home/vcap/app/vendor/bundle/ruby/1.9.1/bin:/home/vcap/app/bin:/usr/bin</td>
</tr>
<tr>
<td>PORT</td>
<td>61001</td>
</tr>
<tr>
<td>PWD</td>
<td>/home/vcap/app</td>
</tr>
<tr>
<td>RACK_ENV</td>
<td>production</td>
</tr>
<tr>
<td>RUBYOPT</td>
<td>-I/home/vcap/app/vendor/bundle/ruby/1.9.1/gems/bundler-1.3.2/lib -rbundler/setup</td>
</tr>
<tr>
<td>SHELL</td>
<td>1</td>
</tr>
<tr>
<td>TMPDIR</td>
<td>/home/vcap/tmp</td>
</tr>
<tr>
<td>USER</td>
<td>vcap</td>
</tr>
</tbody>
</table>

```json
{
    "limits": {
        "mem": 188,
        "disk": 1024,
        "fds": 16384
    },
    "application_version": "d10c4dc6-301c-4b10-9d1f-87b4f619c69d",
    "application_name": "cf-env",
    "application.uris": [
        "env.10.244.0.34.xip.io"
    ],
    "version": "d10c4dc6-301c-4b10-9d1f-87b4f619c69d",
    "name": "cf-env",
    "space_name": "test",
    "space_id": "227de84a-df41-48e1-950c-255e38b2e8f9",
    "uris": [
        "env.10.244.0.34.xip.io"
    ],
    "users": null,
    "instance_id": "219721499d694ac9b7aa4fa113aee84b",
    "instance_index": 0,
    "host": "0.0.0.0",
    "port": 61001,
    "started_at": "2014-06-25 04:44:57 +0000",
    "started_at_timestamp": 1403671497,
    "start": "2014-06-25 04:44:57 +0000",
    "state_timestamp": 1403671497
}
```
I DON'T ALWAYS DEPLOY CLOUD FOUNDRY

BUT WHEN I DO, IT'S WITH BOSH-LITE
Architecture and Operations

• Install Prerequisites
• Install Cloud Foundry Runtime on BOSH Lite
• Cloud Foundry Architecture Overview
• Cloud Foundry BOSH Overview
• Test Your New Private Cloud Foundry!
Coming Up

• Lunch (12:15 - 1:30)

• Part Three (1:30 - 3:00)
  • Learn How CF Enables Continuous Delivery
  • Setup a CD Pipeline with Jenkins, Artifactory, CF Client, and PWS

• Part Four (3:15 - 4:45)
  • Learn CF Extension Points
  • Customize a Buildpack -OR-
  • Write and Deploy a Service Broker
THANK YOU!

See you after lunch!

Matt Stine (@mstine)
Cloud Foundry Platform Engineer
matt.stine@gmail.com
http://www.mattstine.com
Credits

Thanks to Cornelia Davis (@cdavisafc) for contributing the animated architecture portion of this slide deck.