Intro to Cloud Foundry

Tim Harris
Director of Certification, Cloud Foundry Foundation
Introductions

Cloud Foundry Foundation
- 5.0.1 (c) 6, Not for profit
- Supports community and code
- Part of Linux Foundation

Speaker: Tim Harris
- PhD in Computer Science
- Time at Oracle and VMware
- Training and Cert at Cloud Foundry Foundation
Agenda

Intro to PAAS
Why Cloud Foundry?
Developing Applications on CF
Cloud Native in CF
Services and Routes
Multi-Cloud and Heterogeneous Platforms with CF
Platform as a Service

Enterprise IT (legacy IT)
- Applications
- Security
- Databases
- Operating Systems
- Virtualization
- Servers
- Storage
- Networking
- Data Centers

Infrastructure (as a Service)
- Applications
- Security
- Databases
- Operating Systems
- Virtualization
- Servers
- Storage
- Networking
- Data Centers

Platform (as a Service)
- Applications
- Security
- Databases
- Operating Systems
- Virtualization
- Servers
- Storage
- Networking
- Data Centers
Platform as a Service

High level of abstraction
  • Broad “opinions” about how things are done best
  • Less work to implement if you align with those assumptions

Increase agility of developers
  • Quickly iterate on production deployment
  • Reduce churn between silos
Dev-Ops Virtuous Cycle

Continuous Business Planning

Collaborative Development

Continuous Monitoring

Continuous User Feedback & Optimization

Continuous Release and Deployment

Continuous Testing

DevOps
Continuous Feedback

Operate

Deploy

Develop

Steer
Dev-ops Context

Cloud Foundry Lives in automated world
• But doesn’t live alone
• E.g. CI/CD toolset has high value

CF Developers less dependent on other teams
• Testing, Staging, Production Deployment, Operations

Iterative improvement vs large releases
• Reduce cost per deployment
• Decrease time to market
Container Orchestration vs PAAS

Container Orchestration (e.g. Kubernetes)
• Lower level of abstraction
• More control and hence more complexity
• Users manage their own containers

PAAS
• Higher level of abstraction
• Less control but likely more productivity
• Platform manages containers (generally)
• Platform manages buildpacks, dependencies, security and routing
Kubernetes and PAAS

Matt Cholick @cholick · Aug 7
Remember reading this in 2017 & not believing: twitter.com/kelseyhightowe…. Since then, having spent many sprints with K8s, learned it's 100% true. Working with @cloudfoundry for 4 years and K8s for ~1: raw K8s has the primitives for building platforms; it's not a tool to deploy apps

Kelsey Hightower @kelseyhightower
Kubernetes is a platform for building platforms. It's a better place to start; not the endgame.
Container Orchestration vs PAAS

**Buildpacks**
- Developer brings app
- Platform provides standard Buildpack for runtime*
- Platform provides fixed OS container image
- Platform provides fixed host OS Kernel

**Docker**
- Developer brings app
- Developer brings runtime Docker image
- Developer brings Docker OS image
- Platform provides fixed host OS Kernel

AND

**Application Container**
Developing Applications on Cloud Foundry
“Here is my source code
Run it on the cloud for me
I do not care how”

Cloud Foundry Haiku
Onsi Fakhouri
cf push
Buildpacks

Multi-language and Extensible
• Bring your own buildpack or use available ones

Automated detection
• Identifies needed dependencies and provides them

User provide app + buildpack = droplet
• Executable container ready to stage and execute
CF Buildpacks and Applications

- Buildpack
- Rootfs
- Stemcell

-> CloudFoundry Container
  - App bits + Middleware
  - Filesystem
  - Kernel & Devices
Command Line Interface (CLI)
Access to Resources in CF

```
$ cf login -a https://api.run.pivotal.io
OK
Targeted org maki-org
Targeted space development

API endpoint: https://api.run.pivotal.io
(API version: 2.51.0)
User: maki@example.com
Org: maki-org
Space: development
```
Multi-tenancy in CF

Multiple tenants share a given CF instance
  • Full isolation of user processes and data

Org typically shared by team of users
  • Each space may belong to a single user

Cloud Foundry has relatively large footprint
  • Proportional to level of services provided
  • CF instances often have over 40 VMs in full deploy
Diego: Container Orchestration for CF Cells
CF Routes

Allocated by default after a CF Push
• Default domain like cfapps.io or cloudfoundry.org

Routes can be fully specified or randomized
• Slack.cloudfoundry.org or slack-unridiculed-checkup.cloudfoundry.org
• Random routes avoid route conflicts

Loadbalancer across instances by default

Routes have lifecycle of their own
• Separate from the apps they route to
Ingress in CF: Routes
User Authentication and Authorization

UAA is built into platform and manages:

- Authentication: Who are you?
- Authorization: What can you access?

Implements Oauth 2.0 Standard
CF Components Review

CF Push Experience
• Using buildpacks to build containers dynamically

Diego application runtime
• Pushes composed droplets to cells for execution
• Manages scaling and availability at runtime

Routes provide ingress by default

Authentication and Authorization via UAA component
Cloud Native in CF
Cloud Native Apps in CF

Monolith vs Micro-services
- Reusable components
- Separation of Concerns

Stateless and Scalable
- Assume apps are ephemeral
- State is kept external to App

Roughly aligned with 12-factor methodology
- 12factor.net
Backing Services in CF
CF Marketplace

Getting services from marketplace in org cloudfoundry-summit-training / space training as tharris@cloudfoundry.org...

OK

<table>
<thead>
<tr>
<th>service</th>
<th>plans</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Greenplum</td>
<td>Free</td>
<td>Greenplum for Pivotal Cloud Foundry</td>
</tr>
<tr>
<td>MongoDB-cc</td>
<td>standard</td>
<td>A simple MongoDB service broker implementation</td>
</tr>
<tr>
<td>Ion</td>
<td>standard</td>
<td>Scales bound applications in response to load</td>
</tr>
<tr>
<td>app-autoscaler</td>
<td></td>
<td>Performance Testing Platform</td>
</tr>
<tr>
<td>ac</td>
<td></td>
<td>Openmix Global Cloud and Data Center Load Balancer</td>
</tr>
<tr>
<td>blazemeter</td>
<td></td>
<td>Highly available MySQL for your Apps. Managed HA RabbitMQ servers in the Cloud</td>
</tr>
<tr>
<td>cedexisopenssl</td>
<td></td>
<td>PostgreSQL as a Service</td>
</tr>
<tr>
<td>elancier</td>
<td></td>
<td>Mobile Synchronization and Cloud Integration</td>
</tr>
<tr>
<td>cleardb</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cloudamqp</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cloudforge</td>
<td></td>
<td></td>
</tr>
<tr>
<td>elephantsql</td>
<td></td>
<td></td>
</tr>
<tr>
<td>gluon</td>
<td></td>
<td></td>
</tr>
<tr>
<td>gluon</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ironworker</td>
<td></td>
<td></td>
</tr>
<tr>
<td>loadimpact</td>
<td></td>
<td></td>
</tr>
<tr>
<td>memcached</td>
<td></td>
<td></td>
</tr>
<tr>
<td>memcached</td>
<td></td>
<td></td>
</tr>
<tr>
<td>metrics-forwarder</td>
<td></td>
<td></td>
</tr>
<tr>
<td>metrics-forwarder</td>
<td></td>
<td></td>
</tr>
<tr>
<td>miab</td>
<td></td>
<td></td>
</tr>
<tr>
<td>newrelic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>p-circuit-breaker-dashboard</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Applications</td>
<td></td>
<td></td>
</tr>
<tr>
<td>p-config-server</td>
<td></td>
<td></td>
</tr>
<tr>
<td>p-service-registry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pns</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Cloud Foundry Services

External to consuming apps
  • Leaving apps stateless and scalable

Self-service provisioning
  • No demands on IT

CF Broker and Marketplace
  • Broker is app that implements Open Source Broker API
  • Marketplace entry has Service name, plan and cost

Service instance life-cycle
  • Create service instance (service/plan)
  • Bind service to app
  • Unbind and delete service
CF Service Broker
Multi-platform and Multi-cloud
BOSH’s role in Cloud Foundry

Most common deployment method
• Bosh release specifies needed VMs and processes
• Kubernetes deploy also possible

Cloud Provider Interface (CPI)
• Shim between Bosh and IAAS provider
• Maximizes portability across cloud

Supports high availability
• Monitors and restarts VMs and processes
Bosh and Cloud Foundry
Cloud Foundry and Multi-cloud

Cloud Foundry has no knowledge or interest in underlying IAAS
- Bosh deploys VMs via CPI

Portable platform runs on many platforms
- AWS, GCP, Azure, Openstack, private cloud

Diversify investment across vendors
- Mix private/public cloud solutions
- Run on all IAAS with no change
PLATFORM CERTIFICATION

- Cloud.gov
- SUSE
- Atos
- Pivotal
- Huawei
- IBM
- SAP
- Swisscom
Cloud Foundry Container Runtime
Container Runtime in CF

Kubernetes based container orchestration
  • Available on integrated platform
Allows different deployment models for different applications
  • All sharing same infrastructure
Deployed via BOSH as with CF
  • Provides day-2 operational support that we expect
THE ECOSYSTEM
Expanding capabilities

CPIs for IaaS
Service Brokers
Integrators
CLOUD FOUNDRY IS USED BY

61% LARGE ENTERPRISES

14% SMALL ENTERPRISES

25% SMALL AND MEDIUM BUSINESSES
Kubernetes Integration Experiments

CF Container Runtime
• Kube as a service

Eirini – Kubernetes as a Application Runtime
• Replacing Diego runtime with Kubernetes

Bosh CPI for Kubernetes
• Deploy containerized CF via Kubernetes

Istio/Envoy for Cloud Foundry
• Service Mesh and Authentication for External Services
• Common tools in Kubernetes Ecosystem
Summary

Increase developer velocity
- Devops automation
- Cloud Native design principles
- Scalable, Securable, Deployable

Leverages best of OSS communities
- Integration to container orchestration
- Separation of concerns in services
- Building integrations to Istio, Envoy, open service broker API, etc