LET’S BUILD TOMORROW TODAY
Application Centric Microservices

Ken Owens, CTO Cisco Intercloud Services

Redhat Summit 2015
Agenda

• Introduction
• Why Application Centric
• Application Deployment Options
• What is Microservices Infrastructure
• How do you enable Microservices in the enterprise, cloud, and multiple clouds?
• Introducing Shipped
• Conclusion
Introduction

• Vision of Intercloud
• Cisco as a Service
• Platform for IoE

Why Application Centric?
Developers are Driving the Market

- Elastic and “Web-Scale”
- Flexible
- Reduced time to market for apps
- Loosely-coupled components
- “Ruthlessly Standardized”
Alignment to Customer Value (Business Outcomes)
➢ Services vs Legos
➢ Product Alignment vs Project Alignment
➢ Fail Fast
➢ Organizational Aspects
➢ Software Defined Disruption
Practical Examples In The Cloud

- Cloud must enable application integration, development, and deployment
- Cloud Native
- Cloud Valid
- Legacy Architecture
Application Deployment Options
- Openstack as a Service (IaaS, IaaS+)
  - Openstack APIs
  - Orchestration
  - BSS

- Marketplace/Marketplace Federation
  - SaaS
  - Abstraction of underlying infrastructure (IaaS)
  - Geo & Operating Model

- Application Enablement
  - Cloud Native
  - Cloud Transformation
Domain model, cloud user perspective

- Swift
- Nova
- Neutron
- Cinder
- Glance
- Ceilometer
- Keystone
  - Provisioning
  - Horizon
  - Heat

- Object
- Server
- Subnet
- Volume
- Image
- Metric
- User
- Tenant
- Domain

- Container
- Metadata
- Port
- Snapshot
- Alarm
- Etcetera.

- Router
- Floating IP
- VPN
- LB

- Etcetera.
Developing Applications in the Cloud

- Cloud must enable application integration, development, and deployment
- Consumers are interested in agility, flexibility, and business outcomes
- How do we support applications on CCS

- Overview of use cases
  - Cloud Native
    - Integrated or Interoperable -> CICD
  - Cloud Valid
    - Lift & Shift or Interoperable -> CICD
  - Legacy Architecture
    - Lift & Shift -> CICD
What is Microservices Infrastructure
Microservices Definition

- Software architecture style
  - complex applications are composed of small, independent processes communicating with each other using language-agnostic APIs.
  - Application services are small, highly decoupled and focus on doing a small task.
- SOAish
- Quick Comparison
Microservice Advantages (Top of Mind)

✓ Scalability
✓ Resilience / fault isolation
✓ Individual service deployment
✓ Small code base with well defined boundaries.
✓ Flexibility to choose best languages and technologies
✓ Independent development, build and deployment cycle of each Microservice
✓ Enables faster features iteration
✓ Less resistance path to adopt newer technology in future
Micro Services Infrastructure

Single Datacenter

Multiple Datacenter

WAN SERVICE DISCOVERY

dc1.consul

dc2.consul

dc3.consul

© 2015 Cisco and/or its affiliates. All rights reserved. Cisco Public
Microservices Deployment Layout

Support Namespace aware (Secure Isolation)
- Deployments
- Service Discovery and Wiring.
- Load-Balancing
Service Discovery & Load Balancer

1. Registrar monitors Docker events
2. Registrar adds docker instances host:port to service-name mapping to consul.
3. Consul exposes information via inbuilt DNS
4. Consult templates watches changes in Consul
5. HAProxy configuration is updated based on changes in consul
6. Namespace naming convention environment.project.service.shipped.com

Deploy Env. (Staging)

- MicroService-1 (Github repo)
- MicroService-2 (Github repo)
- Dependency-1 (Marketplace App)
- Dependency-2 (Marketplace App)
- Docker containers Mesos – Marathon (MS-infra)

```
1. Registrar monitors Docker events
2. Registrar adds docker instances host:port to service-name mapping to consul.
3. Consul exposes information via inbuilt DNS
4. Consult templates watches changes in Consul
5. HAProxy configuration is updated based on changes in consul
6. Namespace naming convention environment.project.service.shipped.com
```
• Partnering with leading DevOps tool providers:
  • HashiCorp
  • Mesosphere
  • OpenShift

• Intense focus on application and developer centric Service Design

• Exciting new community projects – Build with us: PoC and Contribute:
  • Project Shipped
  • microservices-infrastructure
  • Container networking
  • OpenStack Congress Application Intent (Policy)
  • https://github.com/CiscoCloud/microservices-infrastructure
  • http://developer.cisco.com/Shipped
How do you enable Microservices in the enterprise, cloud, and multiple clouds?
Not so easily…

• Organization
• Process
  • It takes weeks to create a development environment? Too much cost, red tape, politics
  • Non-prod environments are so different than prod?
  • Tests aren’t always accurate
  • Versioning, updates are way too difficult
  • Developers can’t get easy access to backend services (databases, security, etc.)
  • Hard it is to build new and innovative apps at #dayjob?
• Software Defined Challenges
Software Defined Developer Challenges

Develop
- Run multiple services.

Build
- SDLC admin control CI

Deploy
- Increased deployment complexity.
• Develop Private and Build and Deploy Everywhere – No Lock-in/No Compromise

• Build through CI/CD flow designed for multi-cloud and “bring your own” with consistent packaging and versioning

• Easy to deploy with service discovery and automatic service availability

• Manage you application and all services from a single interface across private and multi-cloud environments
Introducing Shipped
Automated Deployment of OpenShift from Cisco Marketplace

Soon
Project Shipped and OpenShift Integration.

Q3
PaaS as first-rate Intercloud Citizen. Vs. Tenant VM’s.
OpenShift In Cisco Marketplace.

Automated click to deploy. VM's directly into Tenants’ Project. Initially ‘Bring your own License’. Working with RH to streamline licensing. Working with vendors to integrate value.
• OpenShift fully supported as a deployment target.
• Unified Development / Deployment Environment.
• Use the PaaS layer you are comfortable with.
• Shipped <> PaaS interface is Docker.

Environment Agnostic (Docker ‘under the hood’)
Conclusion
• Application Centric
• Composable Applications/Microservices
• Platform for IoE = Project Shipped
Software-Defined Distribution = Project Shipped

- Experience Project Shipped @ Cisco Live!
  - Hands on - Hackathon
  - Use the product
  - Meet the entire engineering team
  - Get free GitHub and Bintray private repos
Thank you
TOMORROW starts here.