Deploy a Hybrid, Multi-Cloud Container Environment in Less than an Hour

Blue Thunder Somogyi
Solution Architect, AS Software Innovation
Questions?
Use Cisco Spark to communicate with the speaker after the session

How
1. Find this session in the Cisco Live Mobile App
2. Click “Join the Discussion”
3. Install Spark or go directly to the space
4. Enter messages/questions in the space

cs.co/ciscolivebot#BRKCLD-2235
Agenda

• Introduction
• CloudCenter Cloud Management
• Kubernetes Production Deployment
• Docker EE / Swarm Deployment
• Contiv Container Networking
• Application Architecture
• Conclusion
Introduction

What’s the Point?

• Cloud-Native – Enabling **agility for application** deployments
  • Blue / Green deployments
  • Continuous deployment
  • Decoupling microservice development cycles

• Auto-scaling for Container Infrastructure **Independent of Cloud** Tooling

• Ability to **redeploy applications seamlessly** means ability to shift location or target cloud with equivalent ease
  • Leverage automation to arbitrage cloud price changes
  • **Easily scale** to new geographies based on **localized demand**

• What is ‘Container Infrastructure’?
  • The container orchestration stacks such as **Kubernetes, Swarm**, or Mesosphere
Introduction

How is it Done?

One Integrated Platform
Lifecycle Management
New and Existing Applications
Introduction

How is it Done?

One Integrated Platform
Lifecycle Management
New and Existing Applications
Demo Kick-Off
Deploying Kubernetes and Swarm to Private and Public Clouds

• Kubernetes Application Profile
  → private cloud (vCenter)
  → public clouds (AWS / GCP*)

• Docker Swarm Application Profile
  → private cloud (vCenter)
  → public clouds (AWS / GCP*)

*GCP = Google Cloud Platform
Demo Concept
Deploying and Consuming Kubernetes from One Platform

Kubernetes Clusters Onboarded as Cloud targets in CloudCenter

CloudCenter

IT Administrator Tenant

Application Developer Tenant

K8 Application Profile

CloudCenter VM or Baremetal Orchestrator

Line of Business Application (Deployed on K8)

Kubernetes Clusters

CloudCenter Kubernetes Orchestrator

Business Application Profile
Cisco’s Container Maturity Model

- Extension of Domain 10 Cloud Maturity Model
- Used to assess container infrastructure and process functional coverage
- Basis of a number of predefined and validated container infrastructure stacks
- Cisco Advanced Services engagement methodology
**Cisco-Docker MTA Service**

- **Modernize Traditional Applications (MTA)**
  - Service to containerize existing candidate applications without code refactoring
  - Provides existing applications with security, density, and management benefits of containerization

---

Existing Application ➔ Convert to container ➔ Modern Infrastructure
Move to cloud or refresh HW ➔ Modern Methodologies
Integrate to CI/CD and automation systems ➔ Modern Applications
Add new services or start peeling off services from monolith code base
Cisco Container Platform

Kubernetes Lifecycle Management
Kubernetes AuthN and AuthZ

---

Cisco Differentiators

- Pre-configured turn-key solution
- Open architecture/open source components
- Networking and security with load balancing
- Enterprise grade container storage with data persistency
- Multiple infrastructure backends support with governance

---

K8s master 1
K8s master n
K8s Node
Overlay Network
Load Balancer

Persistent Storage
External Communication
CCP For HyperFlex
CCP (VMware, Bare Metal)

Storage
External Network
Cisco and Google Open Hybrid Cloud Solution

On Prem/Colo Data Center

- Existing Services
- Apps | Data
- Cloud Apps

Google Cloud

- Google Cloud Platform
- Google Kubernetes Engine

Istio: Hybrid Cloud Service Management

consistent environment

Networking | Security | Private Cloud Infrastructure | Consumption Management
CloudCenter Cloud Application Management
Cloud Center Application Management

CloudCenter Architecture

- CloudCenter Logical Architecture
- One CloudCenter Manager (CCM) and Health Monitor
- One CloudCenter Orchestrator (CCO) and AMQP node per managed cloud region
- Repo Virtual Appliance optional (required if CloudCenter deployed within a private environment with no Internet access)

http://docs.cliqr.com/display/CCD48/Virtual+Appliance+Overview
Cloud Center Application Management

CloudCenter Architecture

- CloudCenter Basic Deployment
- Quick setup in public cloud using cloud template
Cloud Center Application Management

CloudCenter Architecture

- CloudCenter Production Deployment
- High availability for all components

http://docs.cliqr.com/display/CCD48/HA+Virtual+Appliance+Architecture
Cloud Center Application Management

CloudCenter Application Profiles

• Application Profile is the heart of the application centric cloud management within CloudCenter

• Cloud abstracted blueprint for provisioning an application which is portable across all CloudCenter supported cloud environments
Cloud Center Application Management

CloudCenter Deployment Environments

- Deployment Environments allow CloudCenter Tenant Admins to create predefined logical environments with specific parameters and scope, such as:
  - Which clouds are allowed
  - Which cloud account/role should be used
  - Which networks are allowed
- Allows for definition of simple network profiles to limit or streamline the end user experience
Cloud Center Application Management
CloudCenter Application Auto-Scaling

• User defined scaling policy based on following metrics:
  • CPU Utilization
  • Memory Utilization / Free Memory
  • Disk Utilization / Rate
  • Network Utilization / Rate

• Policies have a configurable ’Breach Period’ to suppress overzealous scaling actions

• Scale-in conditions separately configurable per scaling policy
Cloud Center Application Management

CloudCenter Application Upgrades & Migration

• Application Profile Designer can:
  • Create scripts to upgrade from one version of an App Profile to another
  • Build version control and application lifecycle management into App Profile

• Auto and Advanced options to suit application complexity

• Migration app profile plan allows an application deployment to be backed up, and the data restored into a new deployment of the application (in same or different target cloud / environment)
# Cloud Center Application Management

## CloudCenter Supported Cloud Providers

<table>
<thead>
<tr>
<th>Public Cloud</th>
<th>Private Cloud</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon Web Services (AWS)</td>
<td>VMware vSphere/vCenter 5.1 – 6.5</td>
</tr>
<tr>
<td>Google Cloud</td>
<td>VMware vCloud Director 5.1 – 8.1</td>
</tr>
<tr>
<td>VMware vCloud Air Dedicated Cloud</td>
<td>OpenStack (Havana - Mitaka)</td>
</tr>
<tr>
<td>Microsoft Azure Classic</td>
<td>Cisco UCS Director (UCSD)</td>
</tr>
<tr>
<td>Microsoft Azure Resource Manager (RM)</td>
<td>Microsoft Azure Pack (Update Rollup 6 and later)</td>
</tr>
<tr>
<td>Dimension Data (MCP 1.0, legacy, and MCP 2.0)</td>
<td></td>
</tr>
<tr>
<td>IBM Bluemix (SoftLayer)</td>
<td></td>
</tr>
<tr>
<td>Alibaba</td>
<td></td>
</tr>
</tbody>
</table>

[http://docs.cliqr.com/display/CCD48/Public+Clouds](http://docs.cliqr.com/display/CCD48/Public+Clouds)

[http://docs.cliqr.com/display/CCD48/Datacenters+and+Private+Clouds](http://docs.cliqr.com/display/CCD48/Datacenters+and+Private+Clouds)
Kubernetes Production Deployment
Kubernetes Production Deployment

Kubernetes Architecture Overview

- Kubernetes Master
  - Control nodes for cluster
  - Can be dedicated nodes (recommended for production deployments) or also host workloads

- Production deployment considerations
  - Highly available Kube API server
  - Clustered & Persistent Etcd
  - Cloud multi-AZ master cluster (“Ubernetes-lite”)

- Scheduler
- API Server
- Controller Manager
- Kubelet
- User POD
- etcd

K8s - Kubernetes
Kubernetes Definitions

- **Cluster**
  - A set of machines, called nodes, that run containerized applications managed by Kubernetes

- **Controller**
  - A control loop that watches the shared state of the cluster through the apiserver and makes changes attempting to move the current state towards the desired state (e.g. replication controller, endpoints controller, namespace controller, and serviceaccounts controller).

- **Node**
  - A node is a worker machine in Kubernetes (also known as ‘Minion’).

- **Namespace**
  - An abstraction used by Kubernetes to support multiple virtual clusters on the same physical Cluster.

- **Pod**
  - The smallest and simplest Kubernetes object. A Pod represents a set of running containers on your cluster.

- **Deployment**
  - An API object that manages a replicated application. Each replica is represented by a Pod, and the Pods are distributed among the nodes of a cluster.

- **DaemonSet**
  - Ensures a copy of a Pod is running across a set of nodes in a Cluster

- **Reference:** [https://kubernetes.io/docs/reference/glossary](https://kubernetes.io/docs/reference/glossary)
Kubernetes Production Deployment

Deployment Methods

• kubeadm – beta, not for production use
• kops (AWS only)
• kargo (ansible based)
• Distro Specific
  • Ubuntu / Canonical - Conjure Up
    • [https://conjure-up.io/](https://conjure-up.io/)
  • CoreOS
    • [https://coreos.com/kubernetes/docs/latest/](https://coreos.com/kubernetes/docs/latest/)
• Managed K8 Services
  • Google Container Engine (GCE)
  • Platform9 [https://platform9.com](https://platform9.com)
  • Stackpoint.io
Kubernetes Production Deployment

Fun Fact – Self Bootstrapping K8

• Bootcube creates a temporary K8 control plane
  • Used to bootstrap final K8 control plane using containerized versions of K8 services (POD)
  • K8 monitors and heals its own services
  • Recovery process available for hard down scenario

• Current state of affairs
  • Level 1-4 self hosted K8 available
  • Kubelet running as local system service, everything else within self hosted POD

https://github.com/kubernetes/community/blob/master/contributors/design-proposals/self-hosted-kubernetes.md
Kubernetes Production Deployment

K8 Application Profile Parameters

- Tier Scale
  - Etcd – One to five nodes
  - K8 Manager – One to three nodes
  - K8 Worker - One or more nodes

- Addressing
  - Service Cluster IP Range – subnet to be used for K8 container based services (DNS, network control plane, etc). This is overlay, so must be designated and conflict free
  - Service Cluster Gateway – address to be used as L3 gateway for service cluster subnet
  - Kubernetes Cluster CIDR – Range of addresses for K8 pods

- Policies
  - Scaling policy
Kubernetes Production Deployment

Deployment Artifacts (‘…’ = https://github.com/)

• Kubernetes
  • …/coreos/etcd/releases/download/v3.2.11/etcd-v3.2.11-linux-amd64.tar.gz
  • …/kubernetes-release/release/v1.9.0/bin/linux/amd64/kube-apiserver
  • …/kubernetes-release/release/v1.9.0/bin/linux/amd64/kube-controller-manager
  • …/kubernetes-release/release/v1.9.0/bin/linux/amd64/kube-scheduler
  • …/kubernetes-release/release/v1.9.0/bin/linux/amd64/kubectl

• Container Runtime
  • …/containernetworking/plugins/releases/download/v0.6.0/cni-plugins-amd64-v0.6.0.tgz
  • …/kubernetes-incubator/cri-containerd/releases/download/v1.0.0-beta.0/cri-containerd-
    1.0.0-beta.0.linux-amd64.tar.gz

• Certificates
  • Using CloudFlare SSL utilities v1.2.0 - https://pkg.cfssl.org/
Docker Enterprise Edition Deployment
Docker Swarm Production Deployment

Docker Enterprise Edition (EE) Solution Overview
Docker Swarm Production Deployment

Docker Swarm Architecture

• Swarm Clusters
  • Swarm Managers
    • A three-manager swarm tolerates a maximum loss of one manager.
    • A five-manager swarm tolerates a maximum simultaneous loss of two manager nodes.
    • An N manager cluster will tolerate the loss of at most (N-1)/2 managers.
    • Docker recommends a maximum of seven manager nodes for a swarm.
  • One Manager node elected as leader of Swarm cluster – other Managers forward API requests to the leader
• **Important Note**: Adding more managers does NOT mean increased scalability or higher performance. In general, the opposite is true.
Docker Swarm Production Deployment
Docker EE Best Practices - AppArmor

- AppArmor is a Linux kernel security module.
- You define profiles that control access to specific resources such as files and the network.
- You can apply these profiles to applications and containers.
- Override the default container profile (dockerdefault) with \( \text{--security-opt flag} \)
  - $ docker container run \(--rm -it / --security-opt apparmor=custom-profile hello-world\)

```bash
# deny write for all files directly in /proc
deny @PROC/*/ w,
# deny write to files not in /proc/<number>/** or /proc/sys/**
deny @PROC/{/[^1-9y][^0-9s],[^1-9y]}
# deny /proc/sys/
# deny everything
deny @PROC/sys/
deny @PROC/sys/kernel/
deny @PROC/sys/
# deny everything
deny @PROC/mem
deny @PROC/kmem
deny @PROC/kcore
deny mount,
deny /sys/[^f]*/*
deny /sys/fs/[^f]*/*
deny /sys/fs/[c]*
deny /sys/fs/cg/
deny /sys/firmware
# deny /sys/kernel/**
$ docker info
Containers: 1
Running: 1
Paused: 0
Stopped: 0
Images: 1
Server Version: 17.03.1-ce
Security Options:
  apparmor
  seccomp
Profile: default
users
```
Docker Swarm Production Deployment

Docker EE Best Practices - Non-Root Container Execution

• By default root inside container = root outside container

• User namespaces:
  • In the Linux kernel since 3.8
  • Supported in Docker since 1.10

• How they work:
  • Give a container its own isolated set of UIDs and GIDs
  • These isolated UIDs and GIDs inside the container are mapped to non-privileged UIDs and GIDs on the Docker host.

• To use with docker, use --userns-remap flag (with /etc/subuid and /etc/subgid populated)
  • sudo dockerd --userns-remap=default &

```
$ docker container run -v /bin:/host/bin -it --rm alpine sh

# whoami
root

# id
uid=0(root) gid=0(root)

# rm /host/bin/*
```

This will delete all files in the /bin directory on the Docker host! Don't do it!
Docker Swarm Production Deployment

Docker EE Best Practices – Trusted Images

• Docker Content Trust (DCT)
  • Enabled in Universal Control Plane (UCP)
  • When enabled, all images are pulled by digest (≈ object store ID)
  • Ensures only verified/untampered images are pulled
  • Will not execute images that fail digest verification

```
$ docker pull alpine@sha256:3dcdb92...b313626d99b889d0626de158f73a
sha256:3dcdb92d7432d...e158f73a: Pulling from library/alpine
e110a4a17941: Pull complete
Digest: sha256:3dcdb92d7432d56604...47b313626d99b889d0626de158f73a
Status: Downloaded newer image for alpine@sha256:3dc...b889d0626de158f73a
```
Contiv Container Networking
What is Contiv

100% Open Source

The Most Powerful Container Networking Fabric
L2, L3, Overlay or ACI

Rich Policy Model
Introducing Contiv 1.0

What’s New:

- LDAP+ RBAC
- All New User Experience and Workflow
- Kubernetes 1.4+ Support
- Docker 1.12+ Support
- OpenShift 1.4+ Integration
- Simple 1-click Install

Commercially Supported Contiv currently available

- Cisco Advance Services
- Cisco Solutions Support

Open Source at github.com/contiv
Group-Based Policy Framework

Allows Container Isolation

Ability to Provide Granular Micro-Service Security in a Scalable Way
Cisco’s Contiv Container Networking

Supported Contiv Deployments

- Contiv Networking supports both major networking models:
  - The libnetwork CNM model
  - The CoreOS CNI model

- Supported Deployment modes:
  - Layer 2 (VLAN)
  - Layer 3 (Routed / BGP)
  - Overlay (VXLAN)
  - ACI (APIC Integrated)

- Vector Packet Processing (VPP) support planned availability Contiv 2.0 (ETA 1HCY18)
  - https://wiki.fd.io/view/VPP

<table>
<thead>
<tr>
<th>Fabric</th>
<th>Kubernetes</th>
<th>Docker Swarm</th>
<th>OpenShift*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer 2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Layer 3</td>
<td>✓</td>
<td>✓</td>
<td>✓*</td>
</tr>
<tr>
<td>Overlay</td>
<td>✓</td>
<td>✓</td>
<td>✓*</td>
</tr>
<tr>
<td>ACI</td>
<td>✓</td>
<td>✓</td>
<td>✓*</td>
</tr>
</tbody>
</table>

*OpenShift support in beta, fully supported in Contiv 1.1

http://contiv.github.io/documents/networking/
Cisco's Contiv Container Networking

Contiv Deployment and Operations

- Ansible based installer

- Operations through:
  - GUI
    - Tenants, Users, Authorization
    - Nodes, Networks, Policies
    - LDAP
  - CLI
    - $ netctl policy create web-policy
    - $ docker network create --driver contiv ...
  - REST API

Installation is complete
===========================================================================

Please export DOCKER_HOST=tcp://192.168.2.50:2375 in your shell before proceeding
Contiv UI is available at https://192.168.2.50:10000
Please use the first run wizard or configure the setup as follows:
  Configure forwarding mode (optional, default is bridge).
  netctl global set --fwd-mode routing
  Configure ACI mode (optional)
  netctl global set --fabric-mode aci --vlan-range -
  Create a default network
  netctl net create -t default --subnet= default-net
  For example, netctl net create -t default --subnet=20.1.1.0/24 default-net

===========================================================================

$ cd contiv-1.0.3
$ ./install/ansible/install_swarm.sh -f ../cluster/.cfg_legacy-swarm.yml
   -e ${SSH_KEY} -u ${USER} -i
$ cd ..
Cisco’s Contiv Container Networking

Contiv Architecture

• Contiv is made of two major components:
  • Netmaster
    • uses distributed state store like etcd or consul to save all the desire runtime of for Contiv objects.
    • completely stateless, scalable, and restart-able.
    • built heartbeat mechanism, through which it can talk to peer netmasters.
  • Netplugin (Contiv Host Agent)
    • communicate with Netmaster over REST Interface
    • uses json-rpc to distribute endpoints from Netplugin to Netmaster

http://contiv.github.io/documents/gettingStarted/
Traditional (Dev)Operations

• Traditional application development and deployment lifecycle considers the application as a workload that rides on top of the available infrastructure services

• Even in DevOps environments, the delineation between what is Dev team responsibility and Ops team responsibility is often distinct (even if Ops team is ‘developing’ automation and monitoring tools)

• DevOps is still a win, as both sides are speaking the same language
First Generation Cloud-Native

• Early cloud native application architectures required application level libraries to perform various functions such as service discovery, inter-service load balancing, and latency protection
  • See Netflix OSS for examples https://netflix.github.io/
  • Eureka – service registry and LB/failover
  • Ribbon – IPC library with embedded load balancing
  • Hystrix – Latency and fault tolerance framework for IPC
Next Generation Cloud-Native

• Current generation of cloud native infrastructures are incorporating these application policy functions, allowing application code to be unconcerned with topology

• Application architecture and policy are increasingly being abstracted from application code and embedded in infrastructure operations

• Example projects
  • Istio.io, LinkerD
  • Open Policy Agent (OPA)
Application-Infrastructure Virtuous Circle

• Application and Infrastructure architectures are becoming increasingly interdependent

• The capabilities offered by modern cloud-native infrastructure reduce the burden on application developers, and ‘operationalize’ policy rather than codify it

• Application and Infrastructure designs must be increasingly be created and iterated in tandem to optimize the application and operations design characteristics
Demo Results
Demo Results
Docker Swarm and Kubernetes Deployments
Conclusion
Conclusion
Streamlined, Repeatable Container Infrastructure Deployments

- Manage your container infrastructure using the same solution as your business application deployments.

- Kubernetes and Docker CloudCenter Application Profile Downloads
  https://github.com/btsomogyi/cloudcenter-k8
  https://github.com/btsomogyi/cloudcenter-dee

- CloudCenter – more info
  http://www.cisco.com/go/cloudcenter

- Contiv - more info
  http://contiv.github.io/
Questions?
Use Cisco Spark to communicate with the speaker after the session

How
1. Find this session in the Cisco Live Mobile App
2. Click “Join the Discussion”
3. Install Spark or go directly to the space
4. Enter messages/questions in the space

cs.co/ciscolivebot#BRKCLD-2235
• Please complete your Online Session Evaluations after each session

• Complete 4 Session Evaluations & the Overall Conference Evaluation (available from Thursday) to receive your Cisco Live T-shirt

• All surveys can be completed via the Cisco Live Mobile App or the Communication Stations

Don’t forget: Cisco Live sessions will be available for viewing on-demand after the event at www.ciscodive.com/global/on-demand-library/.
Continue Your Education

• Demos in the Cisco campus
• Walk-in Self-Paced Labs
• Tech Circle
• Meet the Engineer 1:1 meetings
• Related sessions
  • BRKDCN-2390: Kubernetes Container Networking
  • BRKACI-2505: Deploying Kubernetes in the Enterprise with Cisco ACI
  • BRKDCN-2627: Implementing PaaS with Cisco ACI: Cloud Foundry and OpenShift
  • LTRACI-2700: Docker integration with Cisco ACI
Thank you