Cisco Live!
January 29 - February 2, 2018 • Barcelona
Cisco Container Platform: A new solution for on-premises and multi-cloud kubernetes cluster management

Matt Johnson – Technologist, DEVNET
David Lapsley – Director, Container Platform Engineering
Sanjeev Rampal – Principal Engineer, Container Platform
Cisco Spark

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-2085
Agenda

• Containers Today
• Problem Statement
• DEVNET + Cisco Container Platform
• Cisco Container Platform Recap
• Cisco Container Platform Technology Deep Dive
• Google Partnership
• Engineering Q&A
Containers Today

Matt Johnson
DEVNET
@mattdashj
You’re (hopefully) not managing production like this…

```
$ ssh host1
host1# docker run container
$ ssh host2
host2# docker run container
$ ssh host3
host3# docker run container
```
Container orchestration is a must

<table>
<thead>
<tr>
<th>Load Balancing</th>
<th>Container</th>
<th>Container</th>
<th>Container</th>
</tr>
</thead>
<tbody>
<tr>
<td>Container</td>
<td>Health Checks</td>
<td>Log Aggregation / Access</td>
<td>Kubernetes</td>
</tr>
</tbody>
</table>

`kubectl scale deployment <name> --replicas=3`
“Kubernetes is an open-source system for automating deployment, scaling and management of containerized applications that was originally designed by Google and donated to the Cloud Native Computing Foundation.”
Docker Enterprise: Supports Kubernetes

1. The best enterprise container security and management

Docker Enterprise Edition

2. The best container development workflow

Docker Community Edition

3. Compatibility with the Kubernetes and Swarm ecosystems

4. Industry-standard container runtime

containerd

Image source: https://www.docker.com/kubernetes
Mesosphere DC/OS: Supports Kubernetes

Image Source: https://mesosphere.com/blog/kubernetes-dcos/
Cloud Foundry: Supports Kubernetes

Image Source: https://www.cloudfoundry.org/container-runtime/
Problem Statement
Revisiting our perfect world - Orchestrators

- Load Balancing
- Container
- Container
- Container
- Health Checks
- Log Aggregation / Access
- Kubernetes
- Developer API
- Hardware Provisioning & Lifecycle
- Networking (Public IP’s, Bandwidth, Security)
- Infrastructure Automation
- Infrastructure health checks
- Software Upgrades
- End user / team support
- Security Updates
- Integration
- Capacity Ops
- Training
- Ops / IT Staffing
Revisiting our perfect world - Orchestrators

<table>
<thead>
<tr>
<th>Load Balancing</th>
<th>Container</th>
<th>Container</th>
<th>Container</th>
</tr>
</thead>
<tbody>
<tr>
<td>Health Checks</td>
<td>Container</td>
<td>Container</td>
<td>Container</td>
</tr>
<tr>
<td>Kubernetes</td>
<td>Developer API</td>
<td>Developer API</td>
<td>Developer API</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Load Balancing</th>
<th>Container</th>
<th>Container</th>
<th>Container</th>
</tr>
</thead>
<tbody>
<tr>
<td>Health Checks</td>
<td>Container</td>
<td>Container</td>
<td>Container</td>
</tr>
<tr>
<td>Kubernetes</td>
<td>Developer API</td>
<td>Developer API</td>
<td>Developer API</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Load Balancing</th>
<th>Container</th>
<th>Container</th>
<th>Container</th>
</tr>
</thead>
<tbody>
<tr>
<td>Health Checks</td>
<td>Container</td>
<td>Container</td>
<td>Container</td>
</tr>
<tr>
<td>Kubernetes</td>
<td>Developer API</td>
<td>Developer API</td>
<td>Developer API</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HW Provisioning &amp; Lifecycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Networking (Public IP’s, Bandwidth, Security)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Infrastructure Automation</th>
<th>Infrastructure health checks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software Upgrades</td>
<td>End user / team support</td>
</tr>
<tr>
<td>Security Updates</td>
<td>Integration</td>
</tr>
<tr>
<td></td>
<td>Capacity Ops</td>
</tr>
<tr>
<td></td>
<td>Training</td>
</tr>
</tbody>
</table>

| Ops / IT Staffing |
DEVNETS’ Experience
DEVNET

> developer.cisco.com

• Cisco’s developer program. Focusing on all things API.
  • DEVNET works a lot with Customers, partners and external organizations.
  • Through our work we gain an accurate insight into the current trends, pain points and expectations of modern IT; What is and isn’t working on the path to digital transformation.
  • DEVNET also runs its own micro-service based platforms to support our members, from learning labs to sandboxes.
Customer Feedback

• Container-first
  • “Container Management Portal”

• Multi-tenancy
  • Shared platform with resources partitioned by tenant.
  • Strong security integrations.

• Simple operations
  • Lean, simple operational stack.

• Software Defined Networking
  • Micro-segmentation within/without tenants.
  • SDN integration into K8S.

• Upgrades and maintenance
  • Supportable and simple.
  • Decouple CMP lifecycle from computer/storage lifecycle
  • Single, validated process.

• Persistence and State management
  • Supportable and simple solution.
Cisco Container Platform Recap

David Lapsley
Director, Container Platform Engineering
@devlaps
Cisco Container Platform is not “Just Another Kubernetes Distribution”
Cisco Container Platform is a platform for managing Kubernetes clusters and for making it easier and faster to develop, deploy, scale, manage and secure Cloud Native container applications.
## Cisco Container Platform Features

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Turnkey and Open</strong></td>
<td>• Automated installation/updates using virtual appliance</td>
</tr>
<tr>
<td></td>
<td>• Open Source components (e.g. Kubernetes)</td>
</tr>
<tr>
<td></td>
<td>• Fully curated and validated hardware and software stack</td>
</tr>
<tr>
<td></td>
<td>• Programmatic multi-cluster management</td>
</tr>
<tr>
<td></td>
<td>• Target on-premises infrastructure management (IT)</td>
</tr>
<tr>
<td></td>
<td>• Backed by Cisco</td>
</tr>
<tr>
<td><strong>Day-0 and Day-N Lifecycle Management</strong></td>
<td>• Seamless upgrades, updates and scaling</td>
</tr>
<tr>
<td><strong>Premium Networking and Security</strong></td>
<td>• Seamless container networking (Contiv-VPP, Contiv-ACI)</td>
</tr>
<tr>
<td></td>
<td>• Policy based network segmentation</td>
</tr>
<tr>
<td></td>
<td>• Security of “Hard” tenant isolation</td>
</tr>
<tr>
<td></td>
<td>• Role Based Access Control</td>
</tr>
<tr>
<td><strong>Turnkey Storage</strong></td>
<td>• Seamless integration with Hyperflex, hyper-converged storage</td>
</tr>
<tr>
<td><strong>Flexible Deployment</strong></td>
<td>• Turnkey integration with ESX</td>
</tr>
<tr>
<td></td>
<td>• Turnkey integration with Bare Metal*</td>
</tr>
<tr>
<td></td>
<td>• Turnkey integration with KVM*</td>
</tr>
<tr>
<td><strong>Hybrid Cloud Support</strong></td>
<td>• On-premises to public cloud connectivity</td>
</tr>
<tr>
<td></td>
<td>• Integral part of Cisco/ Google hybrid cloud partnership</td>
</tr>
</tbody>
</table>
## Cisco Container Platform Stack

<table>
<thead>
<tr>
<th>Control Plane</th>
<th>Data Plane</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage (Hyperflex)</td>
<td>Hypervisor Layer (Hyperflex/VMW)</td>
</tr>
<tr>
<td>Compute Hardware (UCS)</td>
<td>Networking (Nexus 9K)</td>
</tr>
</tbody>
</table>
Cisco Container Platform Stack

Control Plane
- Automation
  - HX Connect
- Orchestration
  - Cluster/Machine Controllers
- Operations

Data Plane

VM

Storage (Hyperflex)
Hypervisor Layer (Hyperflex/VMW)
Compute Hardware (UCS)
Networking (Nexus 9K)

Control Plane Kubernetes

Kubernetes
Fluentd
Prometheus
Kibana
Hyperflex
Contiv
# Cisco Container Platform Stack

## Control Plane

<table>
<thead>
<tr>
<th>Automation</th>
<th>Orchestration</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMware</td>
<td>Kubernetes</td>
<td>Hypervisor Layer (Hyperflex/VMW)</td>
</tr>
</tbody>
</table>

## Data Plane

<table>
<thead>
<tr>
<th>Cluster 1 Workloads</th>
<th>Cluster 1 Ops</th>
<th>Cluster 2 Workloads</th>
<th>Cluster 2 Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pod</td>
<td>Pod</td>
<td>Pod</td>
<td>Pod</td>
</tr>
</tbody>
</table>

## Storage

- (Hyperflex)

## Compute Hardware

- (UCS)

## Networking

- (Nexus 9K)

---

© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public
Managing kubernetes clusters with Cisco Container Platform
# Cisco Container Platform Stack

## Control Plane

- **Automation**
- **Orchestration**
- **Operations**

- **HX Connect**
- **Cluster/ Machine Controllers**

## Data Plane

- **Cluster 1 Workloads**
  - Pods
  - Pods
  - Pods

- **Cluster 1**
  - Kubernetes
  - Ops

- **Cluster 2 Workloads**
  - Pods
  - Pods
  - Pods

- **Cluster 2**
  - Kubernetes
  - Ops

## Components

<table>
<thead>
<tr>
<th>Storage (Hyperflex)</th>
<th>Hypervisor Layer (Hyperflex/VMW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compute Hardware (UCS)</td>
<td></td>
</tr>
<tr>
<td>Networking (Nexus 9K)</td>
<td></td>
</tr>
</tbody>
</table>

Cisco live!
Interacting with Cisco Container Platform

Control Plane
- Cluster Management API
- Cluster Logging & Monitoring

Data Plane
- Cluster 1 Kubernetes API
- Cluster 1 Logging & Monitoring
- Cluster 2 Kubernetes API
- Cluster 2 Logging & Monitoring

Layers:
- Storage (Hyperflex)
- Hypervisor Layer (Hyperflex/VMW)
- Compute Hardware (UCS)
- Networking (Nexus 9K)

Tools:
- Kubernetes
- Fluentd
- Prometheus
- Kibana
- Hyperflex
- Contiv
Create Stack

Control Plane

- Automation
- Orchestration
  - Cluster/Machine Controllers
- Operations

Data Plane

Control Plane Kubernetes

VM

- Storage (Hyperflex)
- Hypervisor Layer (Hyperflex/VMW)
- Compute Hardware (UCS)
- Networking (Nexus 9K)

Software Tools:
- Kubernetes
- Fluentd
- Prometheus
- Kibana
- Hyperflex
- Contiv
Cisco Container Platform Stack

Control Plane
- Automation
- Orchestration: Cluster/Machine Controllers
- Operations

Data Plane
- VM
- VM
- VM

Storage (Hyperflex)

Hypervisor Layer (Hyperflex/VMW)

Compute Hardware (UCS)

Networking (Nexus 9K)

Kubernetes
Fluentd
Prometheus
Kibana
Hyperflex
Contiv
Cisco Container Platform Stack

Control Plane
- Automation
- Orchestration
- Operations
  - Cluster/Machine Controllers

Data Plane
- Cluster 1 Kubernetes
- VM
- VM
- VM

Storage (Hyperflex)
Hypervisor Layer (Hyperflex/VMW)
Compute Hardware (UCS)
Networking (Nexus 9K)
Creating a cluster with Cisco Container Platform
Cisco Container Platform Technology Deep Dive

Sanjeev Rampal
Principal Engineer
Container Platform Engineering
@sr2357
Outline

- Architecture
- Technology Primers
- Contiv-VXLAN
- Contiv-ACI integration
- Containerized Application Deployment Example
Architecture
Architecture Summary

- Cisco CP + Kubernetes + Extensions + Add-ons packaged in 1 simple appliance
  - Curated & qualified stack supported by Cisco & partners
- ”Manager of Clusters” user & workflow experience
  - “Hard” multi-tenancy model
- Deployment platform:
  - On-premise Cisco UCS/ Hyperflex 3.0 with vSphere virtualization
  - Additional options in future
- Networking:
  - Contiv non-ACI mode over standard L2/ L3 physical networks
  - Contiv ACI mode (ACI-Kubernetes) over Cisco ACI/N9K physical networks
- Persistent Storage for Kubernetes:
  - Persistent Volumes/ Persistent Volume Claims, FlexVolume driver
HX vSphere Cluster

CCP – CP1

PG10 100.1.1.0/28

PG20

K8S-Blue

PG30

K8S-Red

100.1.2.0/24

100.1.3.0/24

Leaf e.g. N93xx

Spine e.g. N95xx

ASR1K or any L3 GW

100.1.x.x

vCenter
Cisco Container Platform Detailed Architecture
Physical topology

Integrated k8s container networking + BM/ VM networking fabric

DC Core

Spine Layer: N9k

Leaf: N9k

External IP network

L3 out

L2 from Contiv OVS to fabric leaf switch via ethernet VPC/ link bond

Nx K8S tenant cluster nodes
Cisco Container Platform HyperFlex Storage Integration

FS Block Diagram
Quick Primer
- Microservices, Kubernetes & Networking
Kubernetes networking primer

- **Pod**: co-located group of containers with shared volumes. Smallest deployable units in Kubernetes.
- **Service**: provides single, stable name and address for a set of pods. Act as basic load balancers.
- **Label**: used to organize and select groups of objects based on key/value pairs.
- **Replication Controller**: ensures a set of pods is always up and available.
End goal: Efficiently running a containerized application

- So what does such an application look like?

Ref. Microservices Patterns
Chris Richardson
Kubernetes Network Policy

• Fine-grained specification of how selections of pods are allowed to communicate with each other and other network endpoints

• Network namespace isolation using defined labels
  • directional: allowed ingress pod-to-pod traffic
  • filters traffic from pods in other projects
  • can specify protocol and ports (e.g. tcp/80)

• In Kubernetes: Project admin controlled, UX enhancements

• Going forward (automated multitenant \(\rightarrow\) network policy, multitenant isolation as default, configure/edit/view policies in UI)
Contiv-VXLAN Integration
Contiv architecture
Cisco Container Platform example with 2 tenant k8s clusters

Port Group 10
100.1.1.0/26

Port Group 20
100.1.2.0/24

Port Group 30
100.1.3.0/24

Ctrl Plane
M + N

“Tenant K8S” ‘Blue’
M
N

“Tenant K8S” ‘Red’
M
N

L3 physical gateways

100.1.0.0/16
Single tenant cluster: Contiv VxLan mode over vSphere
VM1

Pod A1
Eth0/ 10.128.1.1/16

contivVxlanBridge
vx-a
vx-b

To m2
To m3

K8S Ingress
NGINX

Host Linux
SNAT

eth0

100.1.2.5

IaaS Hypervisor (ESXi) & VS/DVS

eth0

eth1

VM2

Pod B2
Eth0/ 10.128.10.3/16

contivh0
contivh1

BRKCLD-2085 © 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public
VM1

Pod A1

Eth0/10.128.1.1/16

contivVxlanBridge

vx-a

To m2

vx-b

To m3

K8S Ingress

NGINX

Host Linux

SNAT

100.1.2.5

IaaS Hypervisor (ESXi) & VS/DVS

eth0

eth0

To public internet

VM2

Pod B2

Eth0/10.128.10.3/16

contivh0

contivh1

100.1.2.5

BRKCLD-2085 © 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public
Service load balancing data path

Client 10.1.1.3

Replica 10.1.1.4:8080

Replica 10.1.1.5:8080

Replica 10.1.1.6:8080

src: 10.1.1.3:40000
dst: 10.103.233.222:80

src: 10.1.1.3:40000
dst: 10.1.1.4:8080

src: 10.1.1.3:40000
dst: 10.1.1.5:8080

src: 10.1.1.3:40000
dst: 10.1.1.6:8080

src: 10.1.1.3:40000
dst: 10.103.233.222:80

vpp# nat44 add load-balancing static mapping protocol tcp external 10.103.233.222:80 local 10.1.1.4:8080 probability 80 local 10.1.1.5:8080 probability 10 local 10.1.1.6:8080 probability 10 out2in-only
vpp# nat44 forwarding enable
vpp# nat44 add load-balancing static mapping protocol tcp external 10.103.233.222:80 local 10.1.1.3:40000 probability 30 local 10.1.1.4:8080 probability 10 local 10.1.1.5:8080 probability 40 local 10.1.1.6:8080 probability 40 out2in-only
vpp# nat44 forwarding enable0
Host-service, LB to the same node

Host 172.30.1.2

Kube-proxy
REQ: LB+DNAT
RESP: SNAT

VPP

TAP

TAP

GbE

GbE

Service: 10.103.233.222:80

Replica 10.1.1.4:8080

Replica 10.1.1.5:8080

Replica 10.1.1.6:8080

src: 172.30.1.2:40000
dst: 10.103.233.222:80

src: 10.1.1.4:8080
dst: 172.30.1.2:40000

src: 10.1.1.5:8080
dst: 172.30.1.2:40000

src: 10.1.1.6:8080
dst: 172.30.1.2:40000

src: 10.1.1.4:8080
dst: 172.30.1.2:40000

src: 10.1.1.4:8080
dst: 172.30.1.2:40000

src: 10.1.1.4:8080
dst: 172.30.1.2:40000

src: 10.1.1.4:8080
dst: 172.30.1.2:40000

src: 10.1.1.4:8080
dst: 172.30.1.2:40000

src: 10.1.1.4:8080
dst: 172.30.1.2:40000

vpp# nat44 add load-balancing static mapping protocol tcp external 10.103.233.222:80
local 10.1.1.4:8080 probability 80 local 10.1.1.5:8080 probability 10 local 10.1.1.6:8080 probability 10
out2in-only
vpp# nat44 forwarding enable

vpp# nat44 add load-balancing static mapping protocol tcp external 10.103.233.222:80
local 10.1.1.4:8080 probability 80 local 10.1.1.5:8080 probability 10 local 10.1.1.6:8080 probability 10
out2in-only
vpp# nat44 forwarding enable
Kubernetes Ingress w/NGINX

External routable IPs

VIP: 125.1.1.10
Ports: 80, 443, 8080

https://café.example.com/tea
https://café.example.com/coffee

NGINX
NGINX controller

Tea pod1
Coffee pod2
Coffee pod3

Tea pod2
Coffee pod1
Tea pod3

Internal container SDN Contiv
Kubernetes Ingress with HA (keepalived)

VIP: 125.1.1.10
Ports: 80, 443, 8080

External routable IPs

https://café.example.com/tea
https://café.example.com/coffee

Internal container SDN Contiv

Tea pod1
Coffee pod2
Coffee pod3

NGINX
NGINX controller

keepalived

Tea pod2
Coffee pod1
Tea pod3

keepalived

NGINX
NGINX controller
Contiv-ACI Integration
• Network policies of Kubernetes supported using standard upstream format but enforced through OpFlex / OVS using APIC Host Protection Profiles

• Kubernetes apps can be moved without modification to/from ACI and non-ACI environments

• Embedded fabric and virtual switch load balancing

• PBR in fabric for external service load balancing

• OVS used for internal service load balancing

• VMM Domain for Kubernetes

• Stats per namespace, deployment, service, pod

• Physical to container correlation
ACI Network Plugin for Kubernetes

Native Security Policy Support

1. Install Kubernetes and ACI plugin
2. Build service definitions and define network policy
3. Deploy and scale clusters
4. (Optional) Create EPGs and contracts
5. (Optional) Annotate deployments to move between EPGs

1. Fabric bring up
2. Create Kubernetes system resources in ACI
3. (Optional) Create EPGs and contracts for use in Kubernetes
4. Monitor and observe network telemetry
Using Network Policy and EPGs

**Cluster Isolation**

Single EPG for entire cluster.
(Default behavior)
No need for any internal contracts.

**Namespace Isolation**

Each namespace is mapped to its own EPG.
Contracts for inter-namespace traffic.

**Deployment Isolation**

Each deployment mapped to an EPG
Contracts tightly control service traffic.
Contiv-ACI Cisco Container Platform Tenant EPGs

Application EPGs

<table>
<thead>
<tr>
<th>Name</th>
<th>Alias</th>
<th>Description</th>
<th>QoS class</th>
<th>Intra EPG Isolation</th>
</tr>
</thead>
<tbody>
<tr>
<td>kube-default</td>
<td></td>
<td></td>
<td>Unspecified</td>
<td>Unenforced</td>
</tr>
<tr>
<td>kube-nodes</td>
<td></td>
<td></td>
<td>Unspecified</td>
<td>Unenforced</td>
</tr>
<tr>
<td>kube-system</td>
<td></td>
<td></td>
<td>Unspecified</td>
<td>Unenforced</td>
</tr>
</tbody>
</table>
Contiv-ACI  Cisco Container Platform VMM domains

The Inventory menu displays the VMs, hypervisors, and virtual switches belonging to the fabric. This menu allows you to configure connectivity policies for virtual machine providers such as VMware (vCenter, vShield), Microsoft (SCVMM), Red Hat, and Open Stack. We recommend that you create access policies (using the Configure an interface, PC, and VPC wizard) before launching the virtual machine management (VMM) domain wizard.
# Contiv-ACI Cisco Container Platform Tenants

<table>
<thead>
<tr>
<th>Name</th>
<th>Alias</th>
<th>Description</th>
<th>Bridge Domains</th>
<th>VRFs</th>
<th>EPGs</th>
</tr>
</thead>
<tbody>
<tr>
<td>common</td>
<td></td>
<td></td>
<td>6</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>hx8</td>
<td></td>
<td></td>
<td>2</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Infra</td>
<td></td>
<td></td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>mgmt</td>
<td></td>
<td></td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>
Containerized Application Deployment Example

Sanjeev Rampal
Principal Engineer
@sr2357
Example apps

Container Load Balancing As a Service
- K8S Ingress + NGINX

http://guestbook.com

https://café.example.com/tea
https://café.example.com/coffee

Guestbook app
- FE
- Redis Master
- Redis Slave

Cafe app
- Tea
- Coffee

Contiv networking

Persistent storage
Example application yaml

....

spec:
  tls:
    - hosts:
      - cafe.example.com
        secretName: cafe-secret
  rules:
    - host: cafe.example.com
  http:
    paths:
      - path: /tea
    backend:
      serviceName: tea-svc
      servicePort: 80
Tenant k8s cluster with Contiv networking

<table>
<thead>
<tr>
<th>NAME</th>
<th>STATUS</th>
<th>AGE</th>
<th>VERSION</th>
</tr>
</thead>
<tbody>
<tr>
<td>k8sm-admin-00eaa229-7356-4b15-bbae-ad018f13e600</td>
<td>Ready</td>
<td>1h</td>
<td>v1.7.5+coreos.0</td>
</tr>
<tr>
<td>k8sm-admin-629f2053-0bb4-49ad-99f9-ffa928de9f7f</td>
<td>Ready</td>
<td>1h</td>
<td>v1.7.5+coreos.0</td>
</tr>
<tr>
<td>k8sm-admin-c75f0d23-d659-4f88-8b85-4c621d5222ea</td>
<td>Ready</td>
<td>1h</td>
<td>v1.7.5+coreos.0</td>
</tr>
<tr>
<td>k8sw-admin-32bc6205-4100-447a-a2f5-e1f241b0bd9b</td>
<td>Ready</td>
<td>1h</td>
<td>v1.7.5+coreos.0</td>
</tr>
<tr>
<td>k8sw-admin-8cf8bc9b-fd9d-4b55-9979-6e7369b0c2dc</td>
<td>Ready</td>
<td>1h</td>
<td>v1.7.5+coreos.0</td>
</tr>
</tbody>
</table>

5-node tenant K8S 1.7.5 cluster
Containerized applications running in their namespaces

Guestbook app in default namespace

Cafe app in café-app namespace
Service within Café app accessed from external public network at https://cafe.example.com/tea

Guestbook app accessed from external public network at http://guestbook.com
Cisco + Google Cloud

Sanjeev Rampal
Principal Engineer
@sr2357
Best of Both Worlds

- Networking and Security
- Private Cloud Infrastructure
- Multicloud Management
- Enterprise Class Sales and Support

- Cloud Services
- Microservices / Containers
- API Gateway for Existing Services
- Developer Community
Cisco and Google Open Hybrid Cloud Solution

Data Center

- On premises Kubernetes
- Private Cloud infrastructure
- Legacy Services
  - Apps | Data

Consistent Kubernetes Environment

Service Mesh

Google Cloud

- Google Kubernetes Engine
- Google Cloud Platform

Networking | Security | Private Cloud Infrastructure | Consumption Management
Engineering Q&A

SPEAKERS
TITLES
@twitters
• 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.ciscolive.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
Thank you