
You make possible

Safely using your Nexus 9000 switches as a Kubernetes cluster

Dr Tim Miller, Virtual CSE DC/Cloud
@broadcaststorm

DEVWKS-2096

Cisco live!
June 9-13, 2019 • San Diego, CA
#CLUS

Cisco Webex Teams

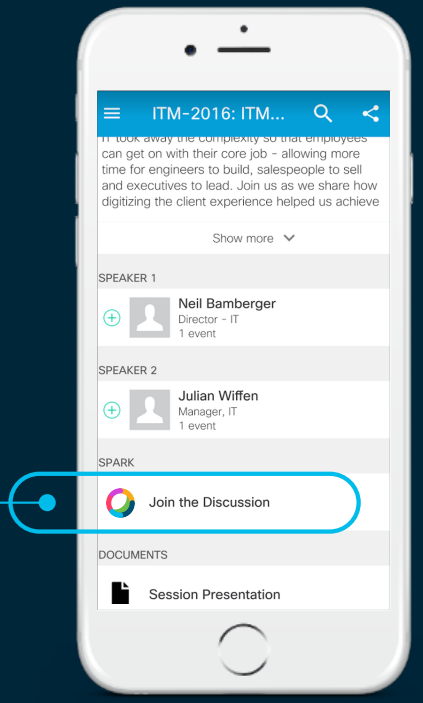
Questions?

Use Cisco Webex Teams to chat
with the speaker after the session

How

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

Webex Teams will be moderated by the speaker until June 16, 2019.



cs.co/ciscolivebot#DEVWKS-2096

Agenda

- Introduction
  - Workshop Goals and Expectation
  - NX-OS Software Architecture
- Deploying Docker on NX-OS
- Deploying Kubernetes on NX-OS
- Deploying Applications on NX-OS Kubernetes clusters
- Conclusion

Introduction

You make customer experience **possible**

Cisco live!

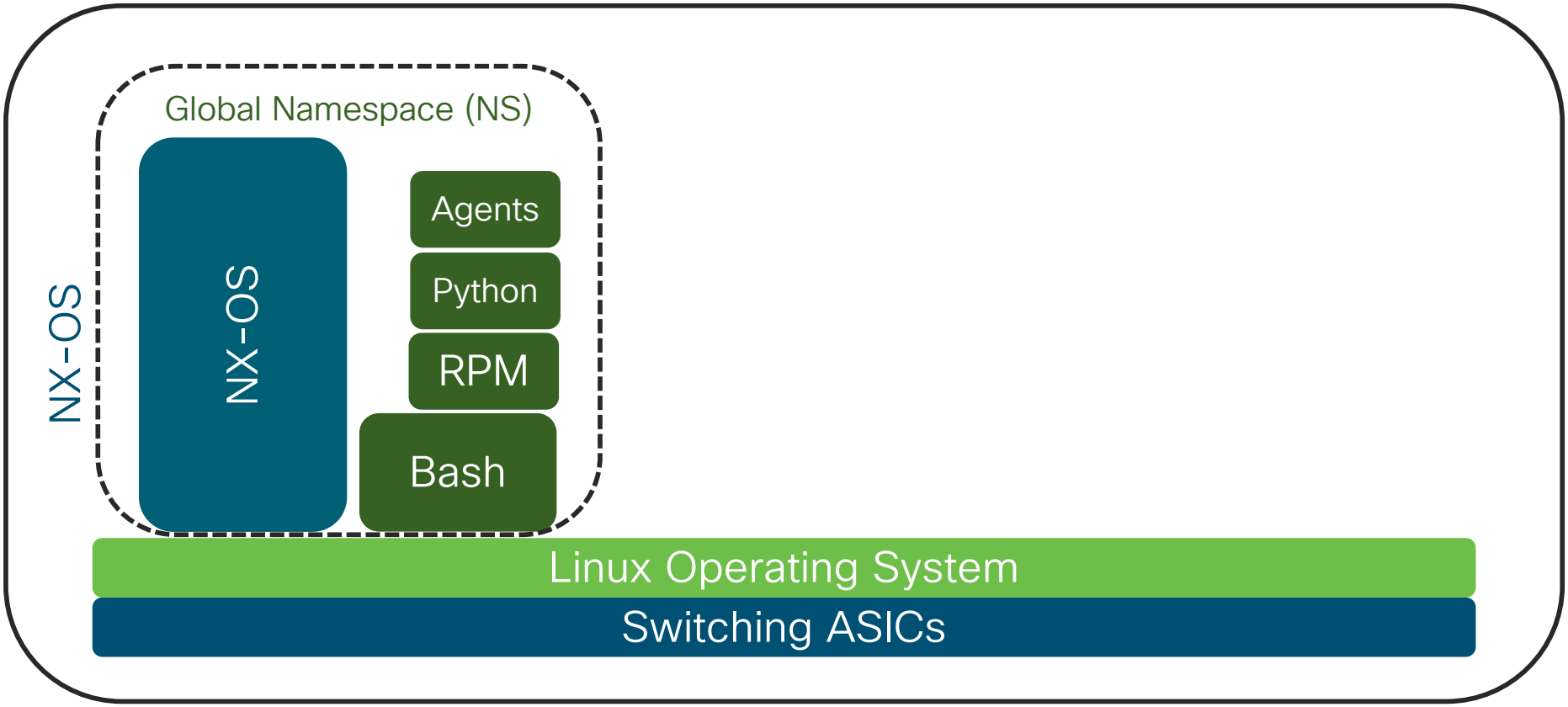
Workshop Goals and Expectations

- The Mission
  - Leverage the spare CPU cycles on your network switches
  - Increase the granularity of your metrics collections
  - Easily deploy applications across the entire network or on a few switches
- The Flight Plan
  - Understand NX-OS and the Linux underpinning it
  - Docker, Kubernetes, Linux namespaces
  - Kubernetes deployment mechanisms
- Destination: Demo Glory

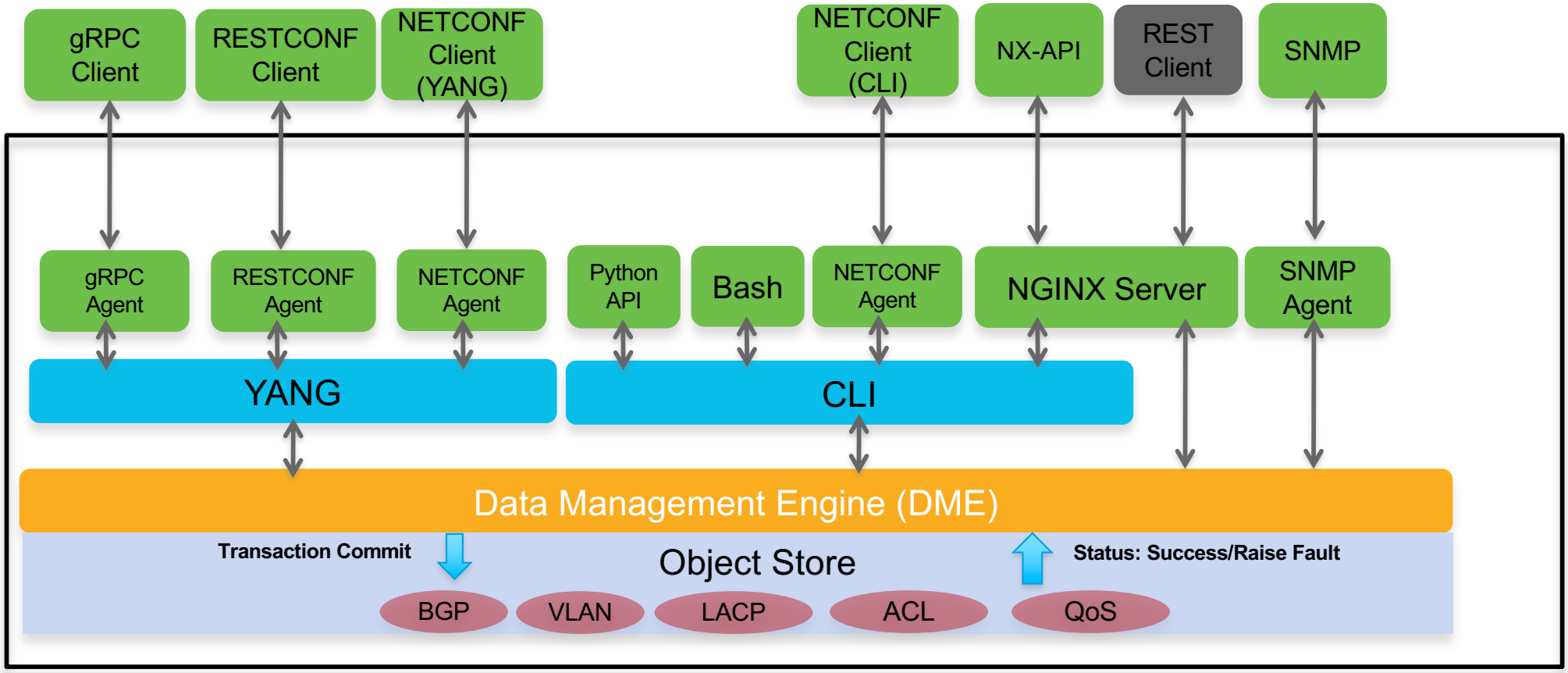
Pre-packed Flight Kit

- Linux – bash, Python, networking (bridges)
- Containers – cgroups and namespaces
- Docker – docker pull/run, image registry
- Kubernetes – terminology (pods, deployments)
- NX-OS – CLI, Switching/Routing (VLANs, VRFs, protocols)

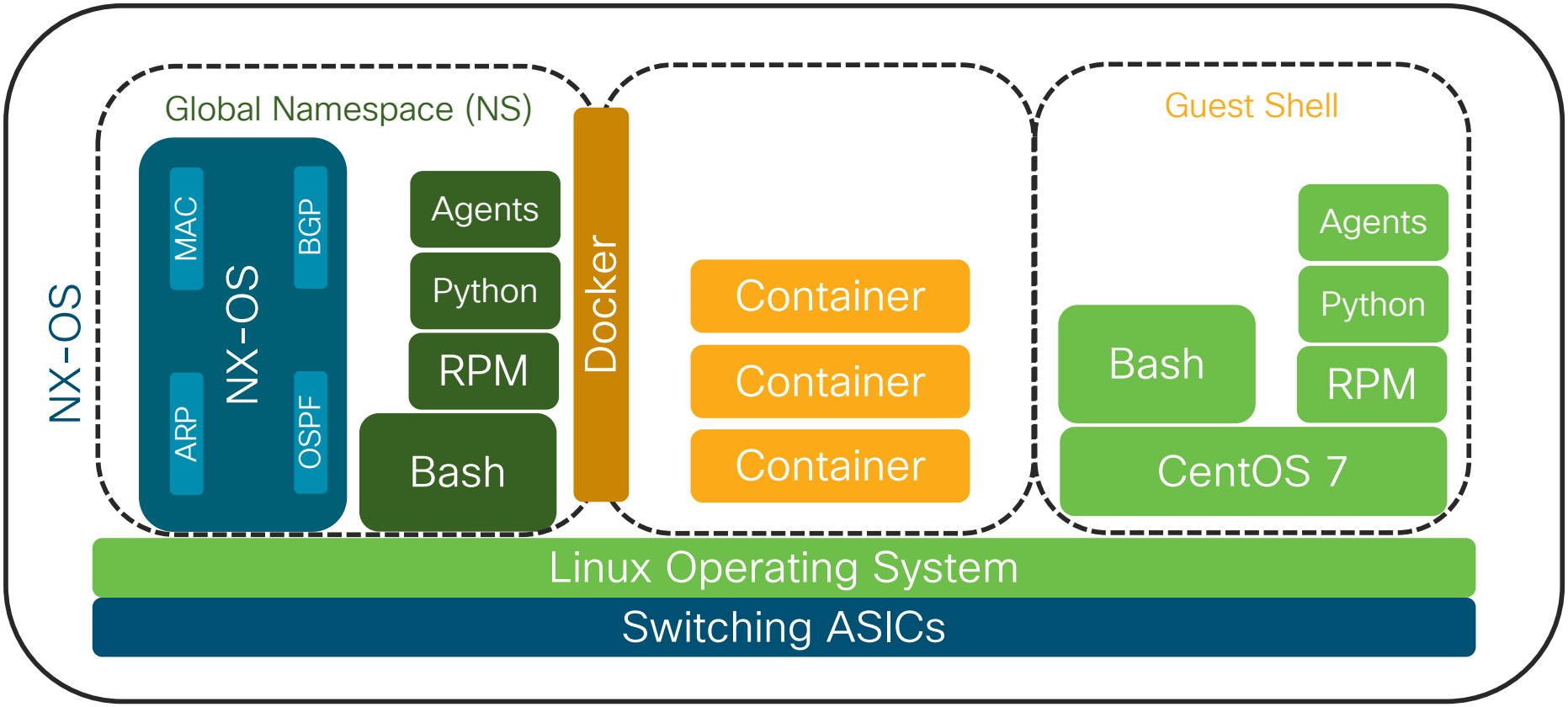
Open NX-OS Architecture



NX-OS Software Architecture



Open NX-OS Architecture

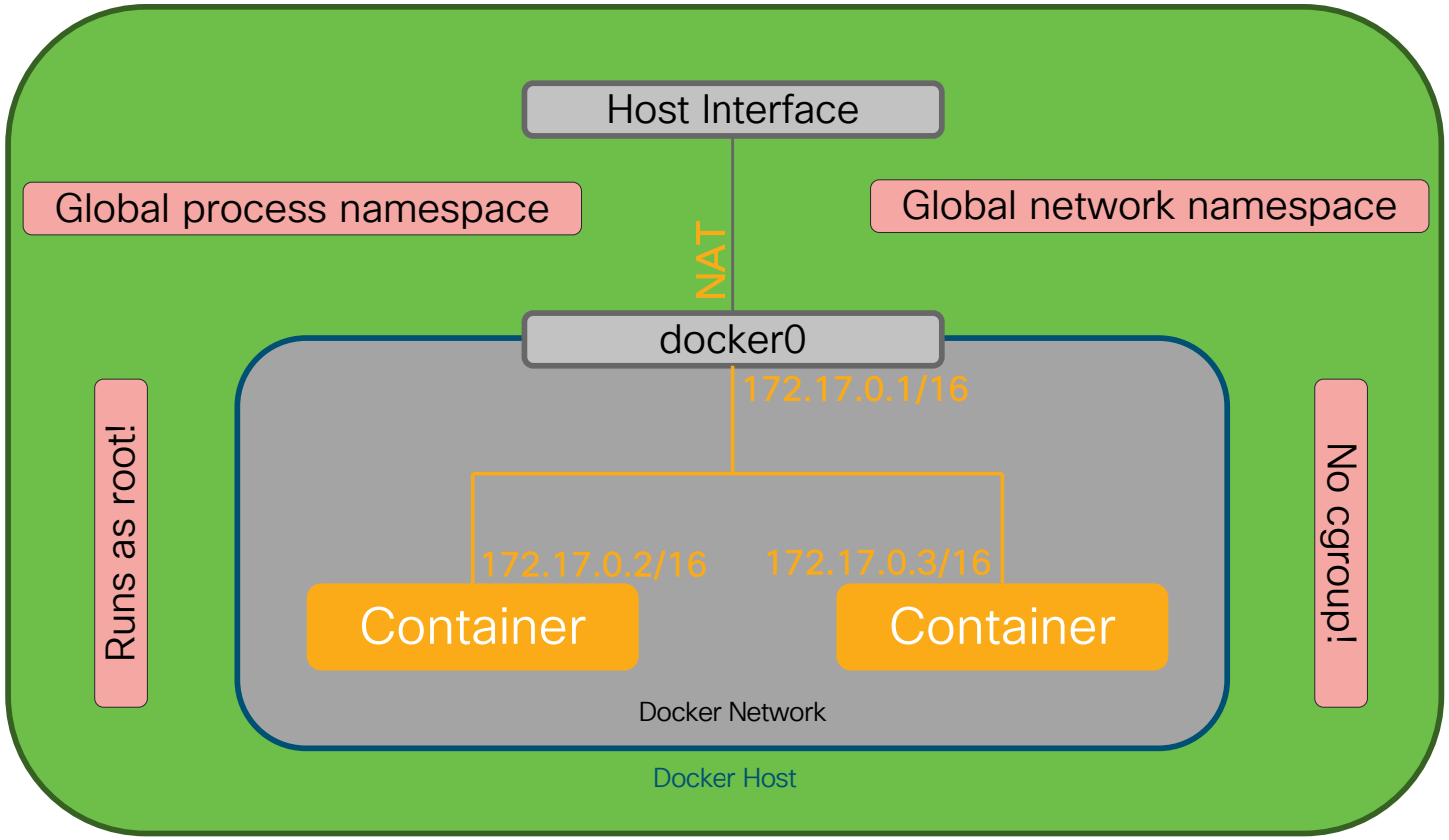


Docker on NX-OS

You make networking **possible**

Cisco live!

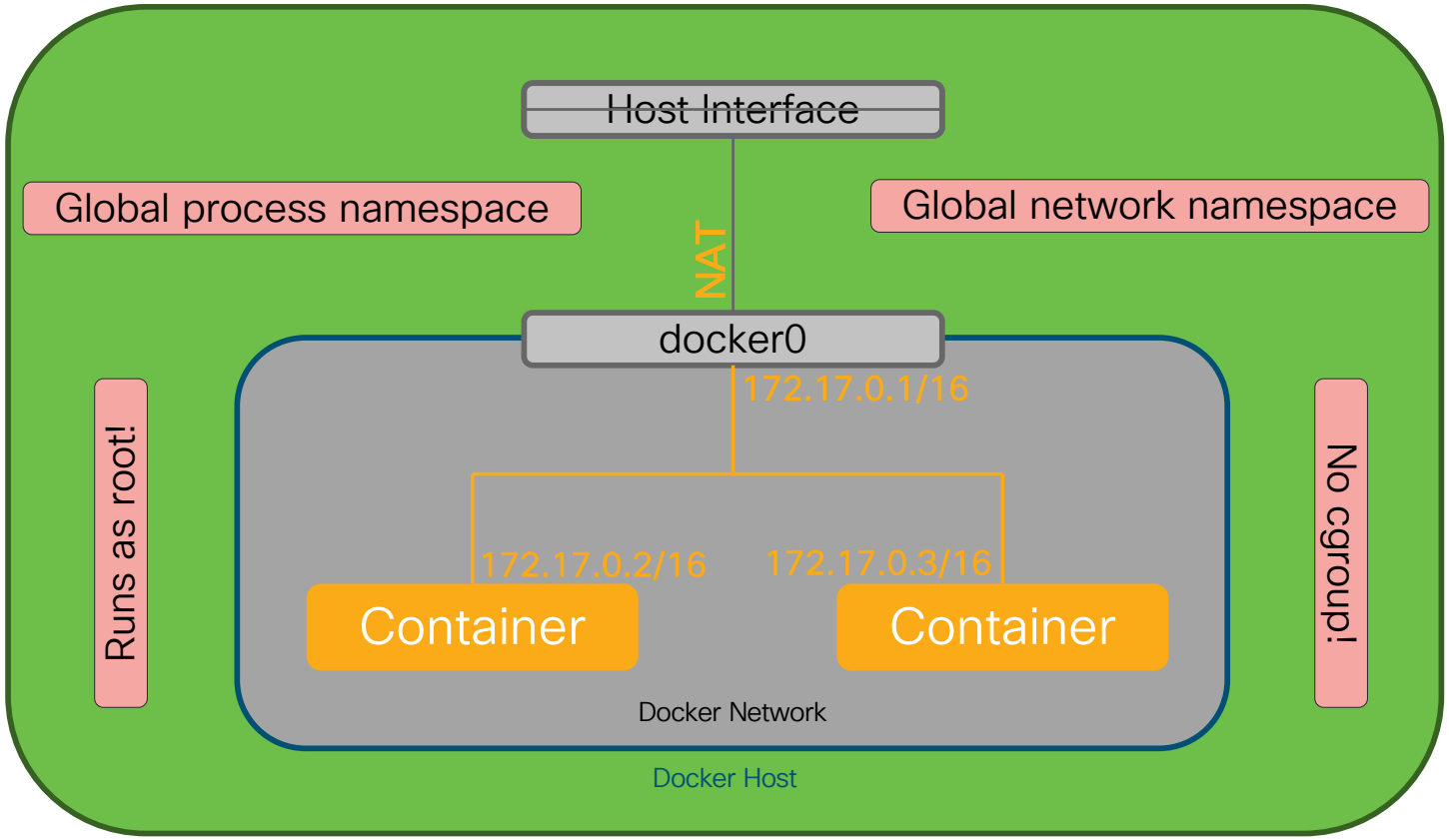
Docker on a server



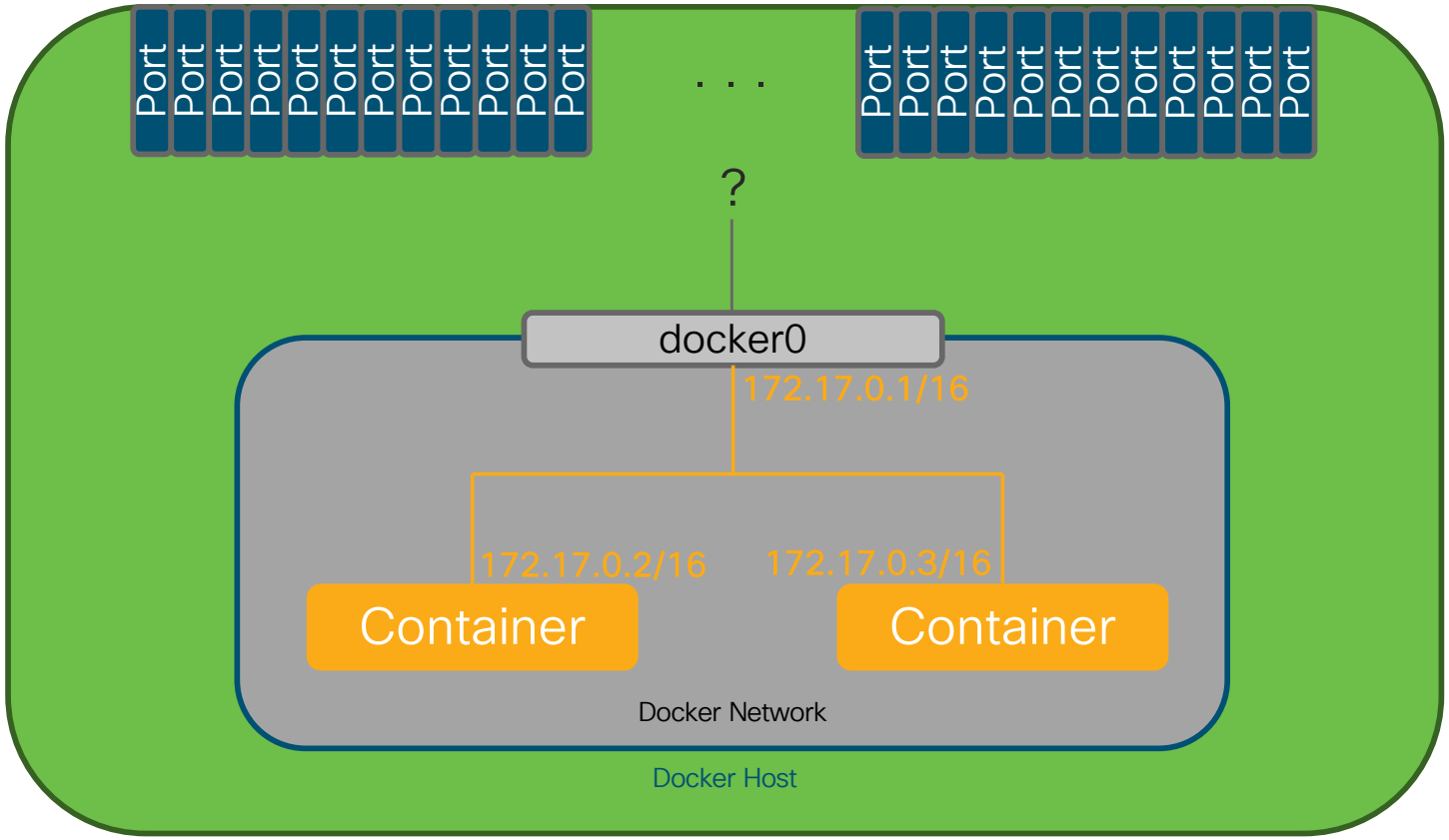
```
$ docker run --name=svr1 -it alpine:latest /bin/sh
/ # ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
2: tunl0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN qlen 1
    link/ipip 0.0.0.0 brd 0.0.0.0
3: ip6tnl0@NONE: <NOARP> mtu 1452 qdisc noop state DOWN qlen 1
    link/tunnel6 00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00 brd 00:00:00:00:00:00:0
0:00:00:00:00:00:00:00:00:00
46: eth0@if47: <BROADCAST,MULTICAST,UP,LOWER_UP,M-DOWN> mtu 1500 qdisc noqueue state UP
    link/ether 02:42:ac:11:00:02 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.2/16 brd 172.17.255.255 scope global eth0
       valid_lft forever preferred_lft forever
/ # ip route show
default via 172.17.0.1 dev eth0
172.17.0.0/16 dev eth0 scope link  src 172.17.0.2
/ # ps auwwx
PID   USER     TIME  COMMAND
    1 root      0:00 /bin/sh
    9 root      0:00 ps auwwx
```

```
$ docker run --name=svr2 -it alpine:latest /bin/sh
/ # ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
2: tunl0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN qlen 1
    link/ipip 0.0.0.0 brd 0.0.0.0
3: ip6tnl0@NONE: <NOARP> mtu 1452 qdisc noop state DOWN qlen 1
    link/tunnel6 00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00 brd 00:00:00:00:00:00:0
0:00:00:00:00:00:00:00:00:00
44: eth0@if45: <BROADCAST,MULTICAST,UP,LOWER_UP,M-DOWN> mtu 1500 qdisc noqueue state UP
    link/ether 02:42:ac:11:00:03 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.3/16 brd 172.17.255.255 scope global eth0
       valid_lft forever preferred_lft forever
/ # ip route show
default via 172.17.0.1 dev eth0
172.17.0.0/16 dev eth0 scope link  src 172.17.0.3
/ # ps auwwx
PID   USER     TIME  COMMAND
    1 root      0:00 /bin/sh
    8 root      0:00 ps auwwx
```

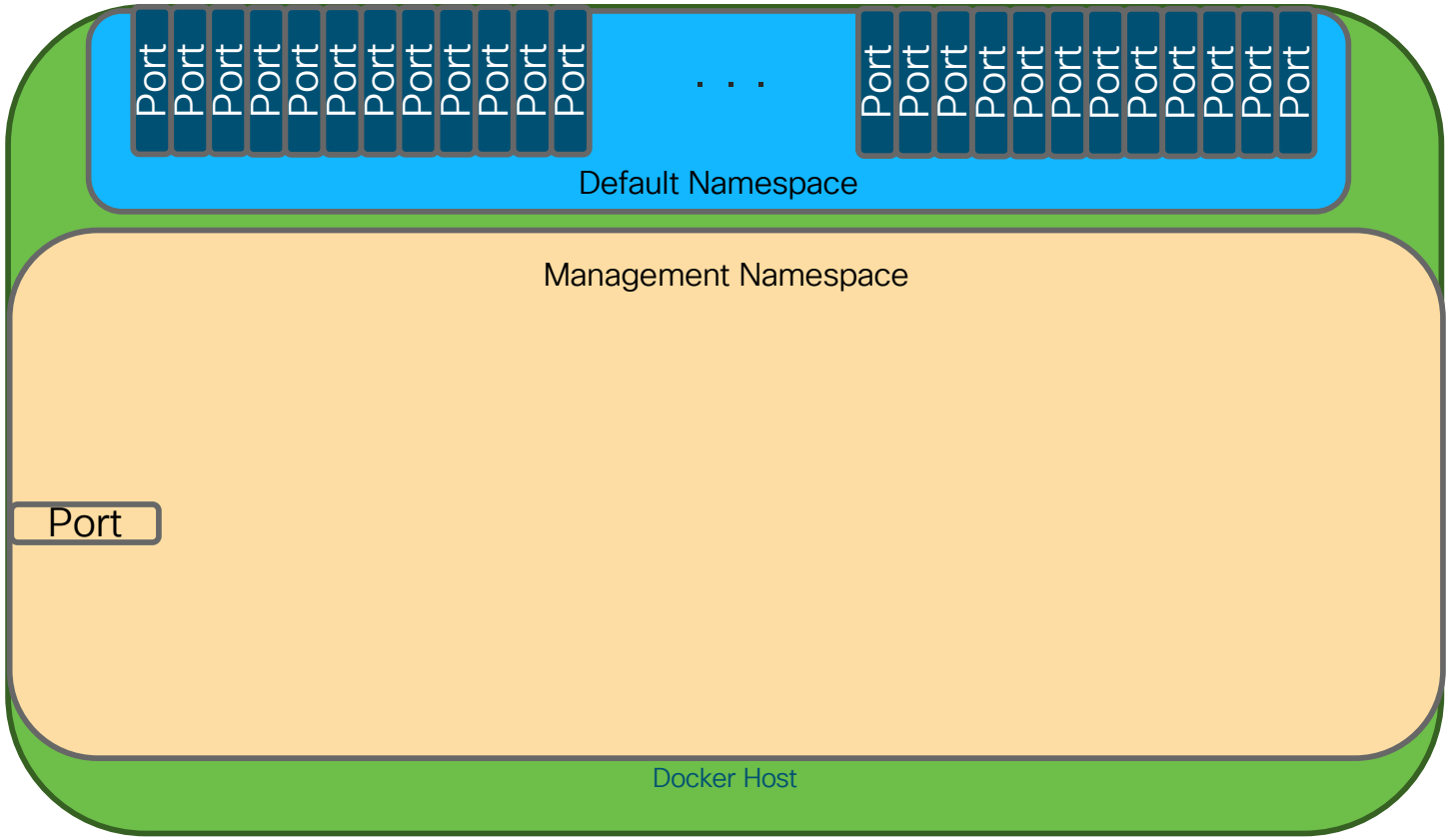
Docker on a server



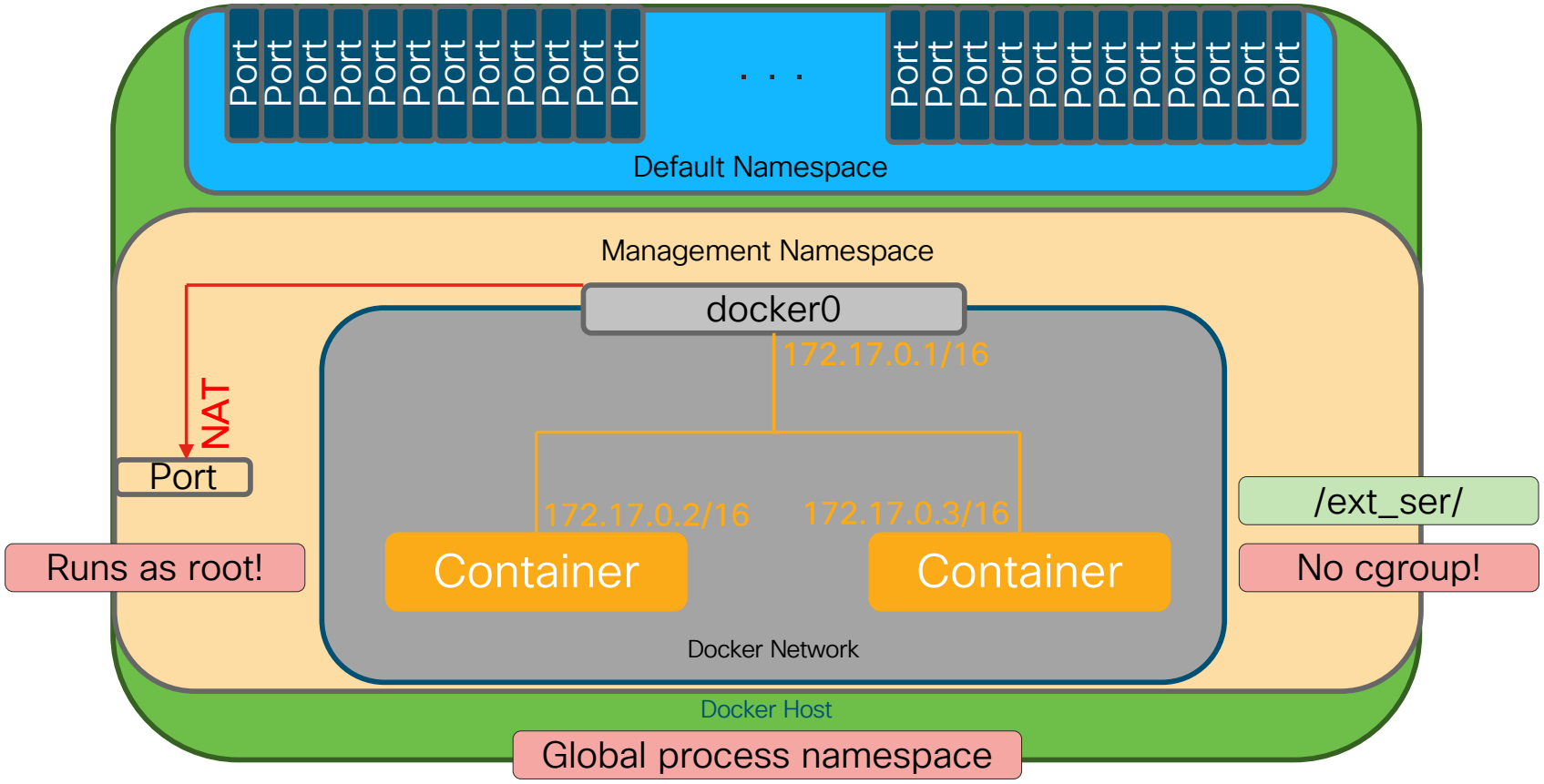
Docker on NX-OS



Namespaces in NX-OS Linux



Docker Networking on NX-OS



Deploy Docker on NX-OS

Cisco live!

Reference Slide

Docker Demo Tasks

- What has been done for you
  - Session Git repo cloned
  - Deploy Vagrant Nexus 9000V instance
  - Configure Nexus 9000v for NXAPI
- What you have to do
  - Use script to configure/setup Docker
  - **vagrant ssh** into Nexus 9000v
  - Two Alpine containers – demonstrate Docker network/process separation
  - Run stress test
  - **vagrant destroy** when you finish

Reference Slide

Docker Deployment Summary

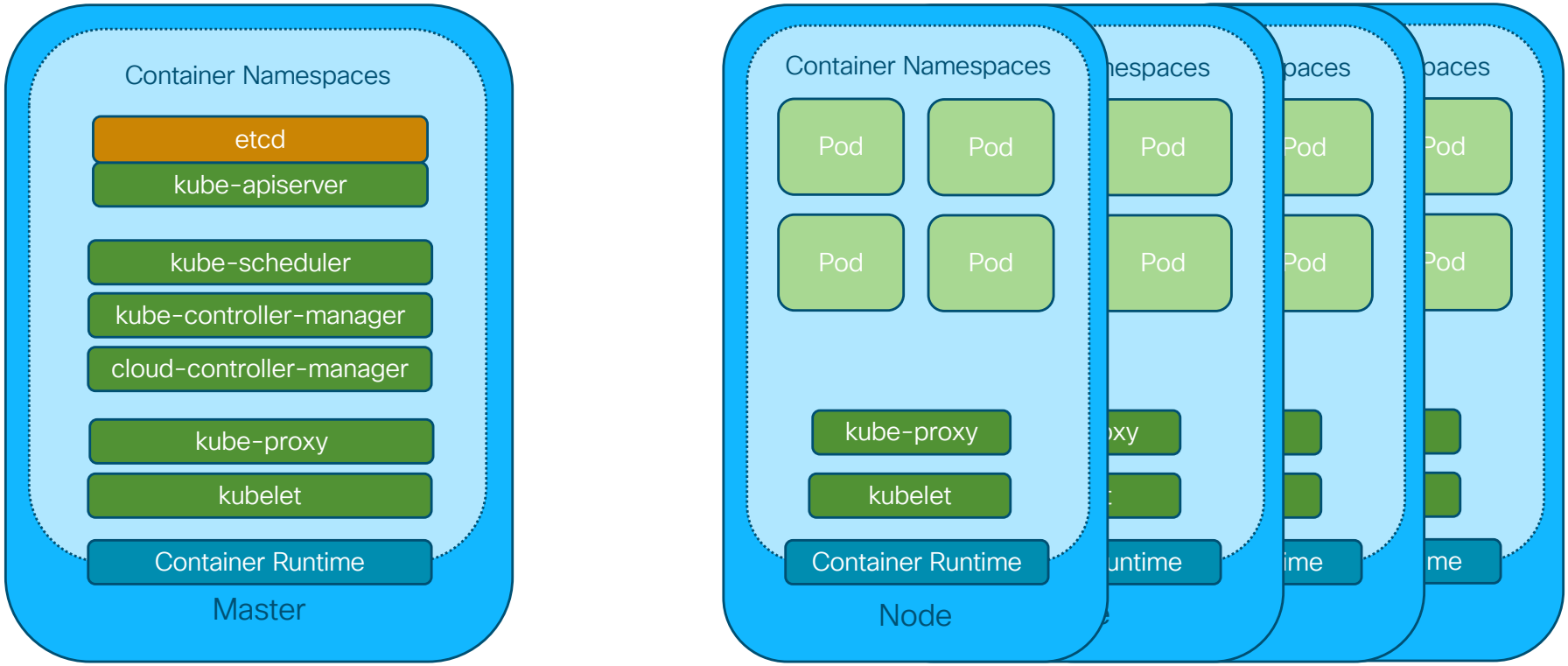
- Remove guest shell (memory, cpu, and storage)
- Initialize Docker
- Re-size container storage
- Secure Docker
- Enable Docker at boot

Kubernetes on NX-OS

You make multi-cloud **possible**

Cisco live!

Kubernetes Cluster Architecture



Kubernetes on NX-OS Architecture

- Master node on external server (VM)
- Worker nodes are Nexus 9000 switches

- Container runtime is Docker (all nodes)
- Kubelet installed as a daemon on all nodes
- All other components are containers running on all nodes

Unique Challenges with NX-OS

- It's Linux. Right?
- Much of Linux environment is stateless
  - /bootflash, /etc (among others) persist
  - /usr/bin, /opt, /var/lib (among others) do not persist
- Fedora or CentOS/RHEL RPMs can't be safely used
  - Requires a bit of "Kubernetes the Hard Way"
- /var/lib/docker is file (/bootflash/dockerpart) mounted via /dev/loop
- /var/lib/kubelet does not exist, needs to persist

https://github.com/kelseyhightower/kubernetes-the-hard-way

Deploy Kubernetes on NX-OS

Reference Slide

Kubernetes Deployment Summary

- Ensure access to Google Container Registry (gcr.io)
- Get your versions correct
- Deploy master node
- Deploy worker nodes

Reference Slide

Kubernetes Deployment Review

- Review the k8s-master-setup.sh
- Create the master
- Review the configuration generation and distribution
- Create the configs and distribute them
- Review the k8s-worker-setup.sh
- Create the workers

Applications on Kubernetes on NX-OS

You make the power of data **possible**

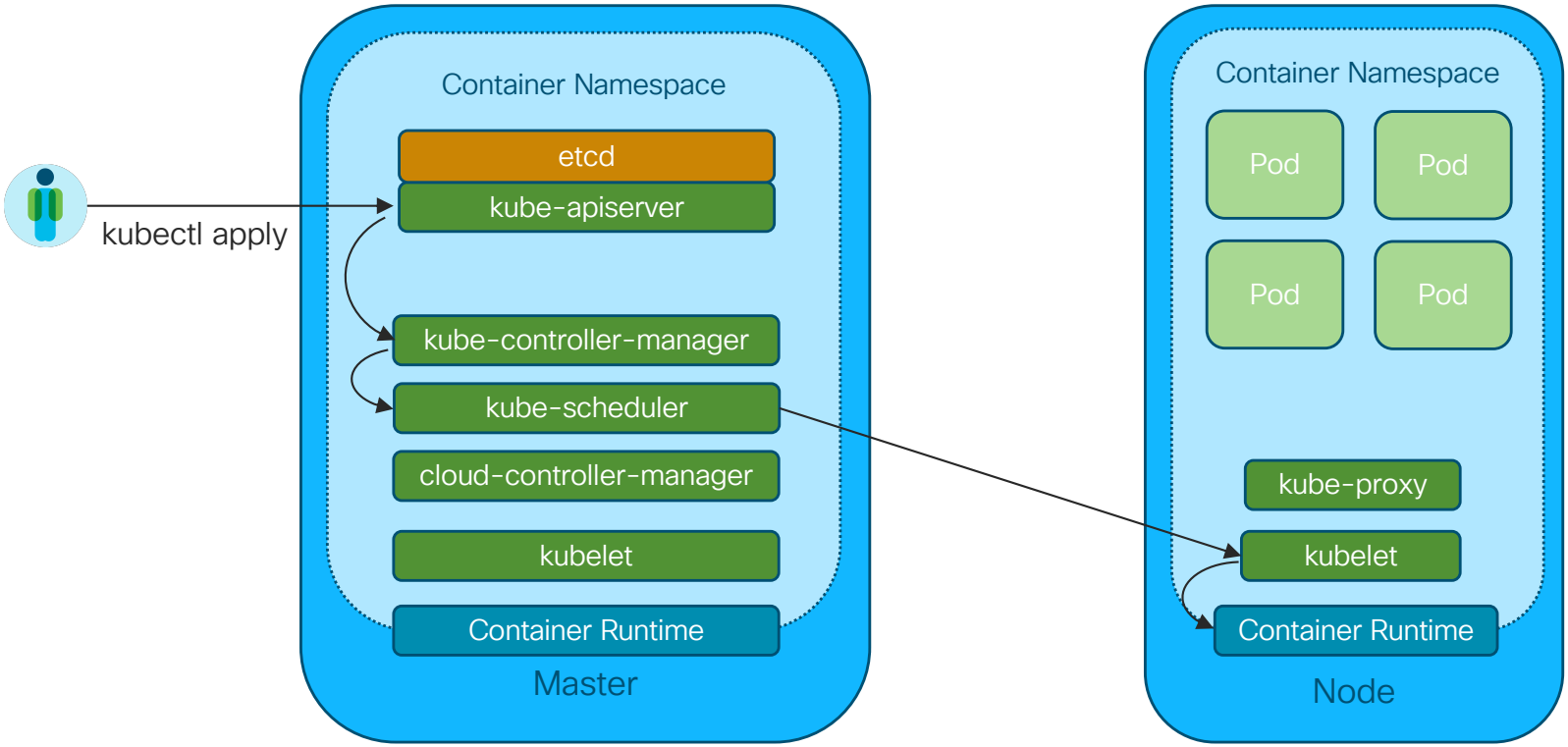
Ciscolive!

Kubernetes Application Terminology

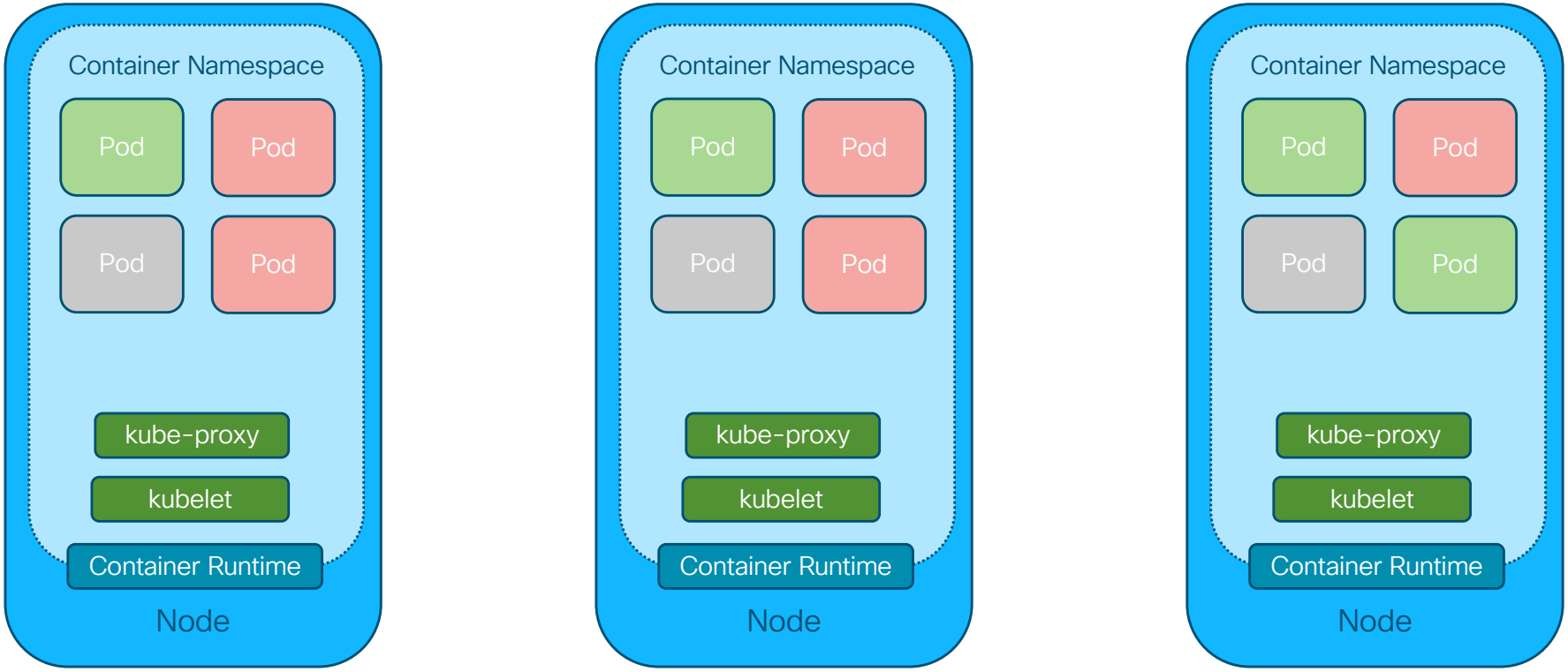
- Pod
  - The atomic unit of deployment. Consists of one or more containers.
- ReplicaSet
  - Resource that ensures 1 or more pods are running
- Deployment
  - Resource that defines a desired state for Pods/ReplicaSets
- DaemonSet
  - Similar to Deployment, except placement is one pod/replica per node

Reference Slide

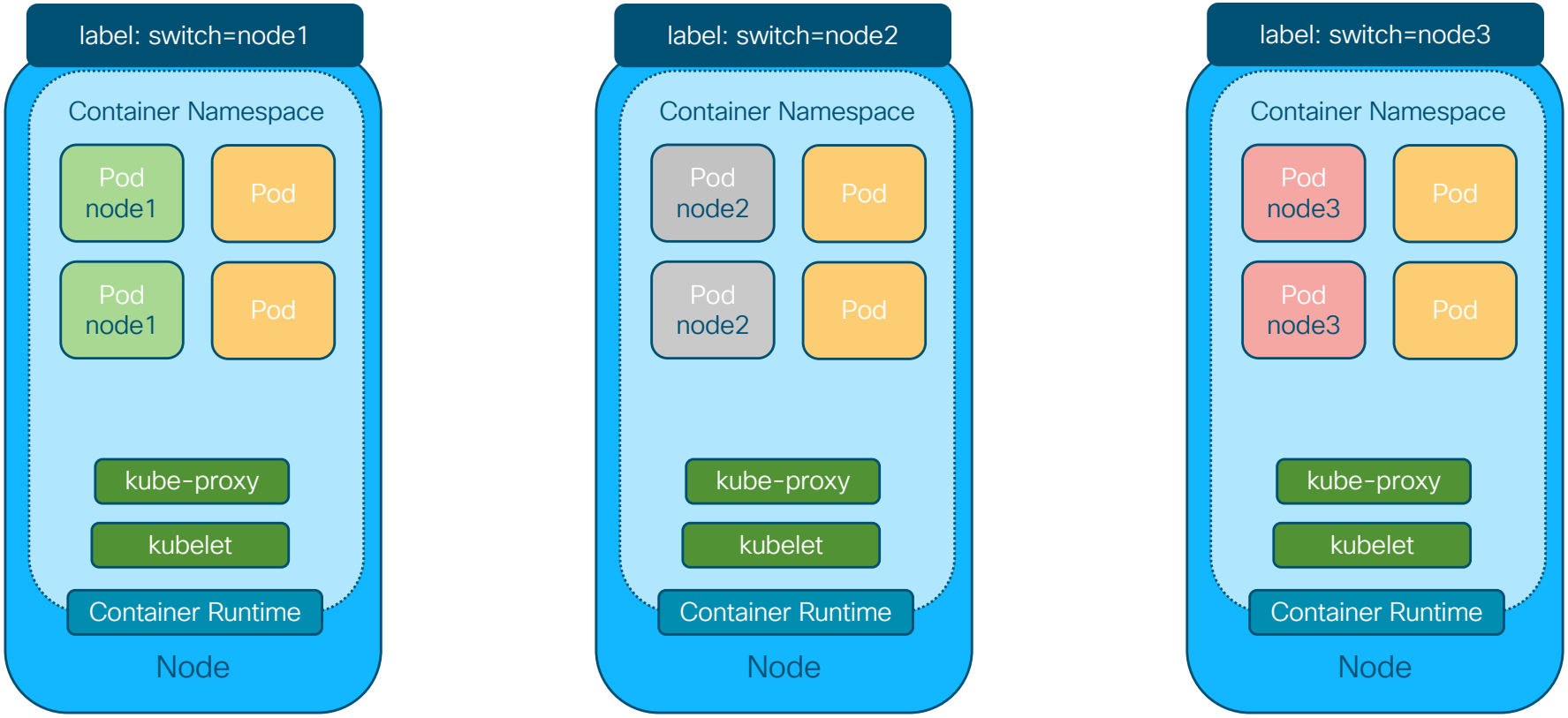
Kubernetes Orchestration



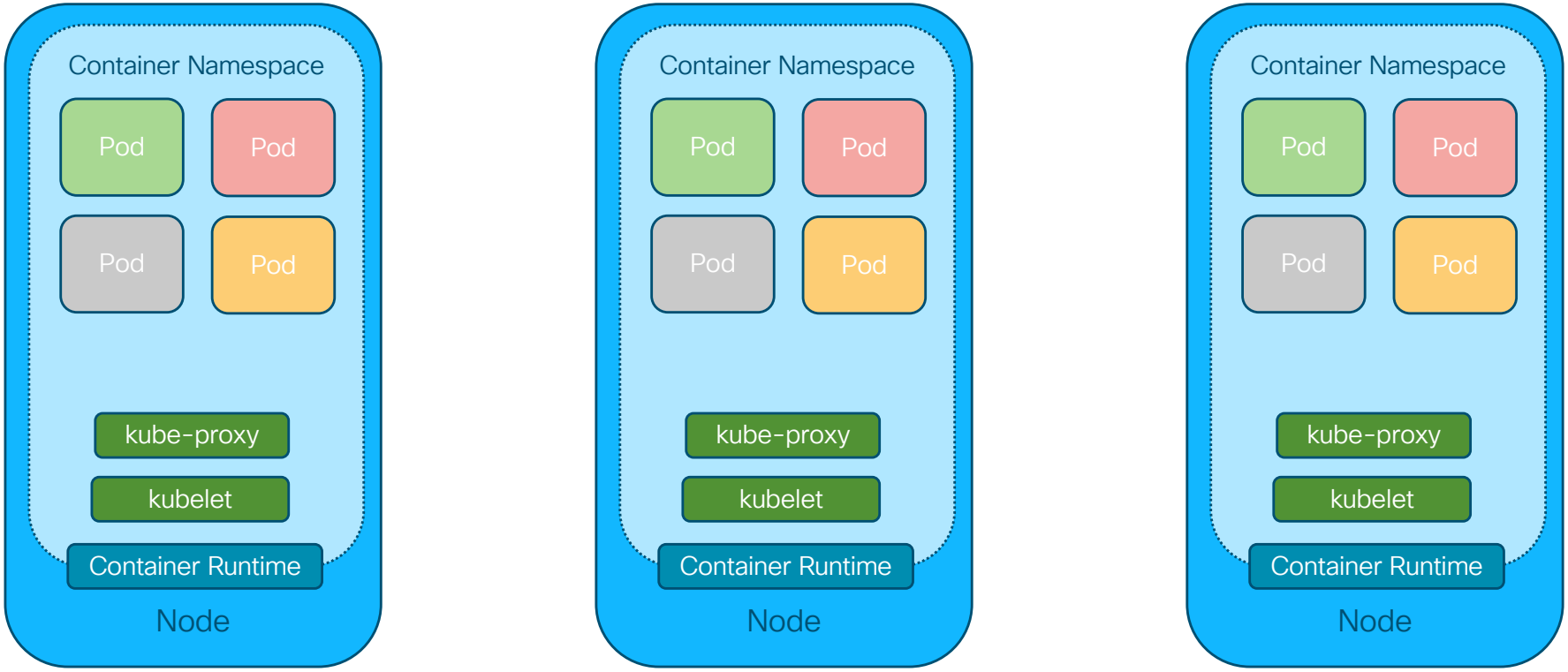
Kubernetes Pod, Replicas, and Deployments



Kubernetes Pod Placement with Labels



Kubernetes Pod Placement with DaemonSet



Deploy Applications on Kubernetes

Application Deployment Summary

- Deploy sample application as:
  - Standard deployment
  - Label selected deployment
  - Daemonset deployment
- Upgrade Daemonset deployment from version 1 to version 2

Current Limitations with NX-OS Kubernetes

- hostNetwork attachment for pods
  - Service IPs and Load Balancing are WIP
- ReplicaSets are fine but upgrades aren't clean

Summary

You make networking **possible**

Summary

- Docker fully supported in NX-OS
  - Lives within management VRF
  - Containers have full access to NXAPI on switch
- Kubernetes for deployment of per-node applications/containers
  - Reduce burden of deploying/scheduling applications
  - Reduce burden of assuring running application
  - Naturally integrate with Kubernetes ecosystem for CI/CD or DevOps
- Leverage Kubernetes natural mechanisms for upgrades

Complete your online session evaluation

- Please complete your session survey after each session. Your feedback is very important.
- Complete a minimum of 4 session surveys and the Overall Conference survey (starting on Thursday) to receive your Cisco Live water bottle.
- All surveys can be taken in the Cisco Live Mobile App or by logging in to the Session Catalog on ciscolive.cisco.com/us.

  Cisco Live sessions will be available for viewing on demand after the event at ciscolive.cisco.com.

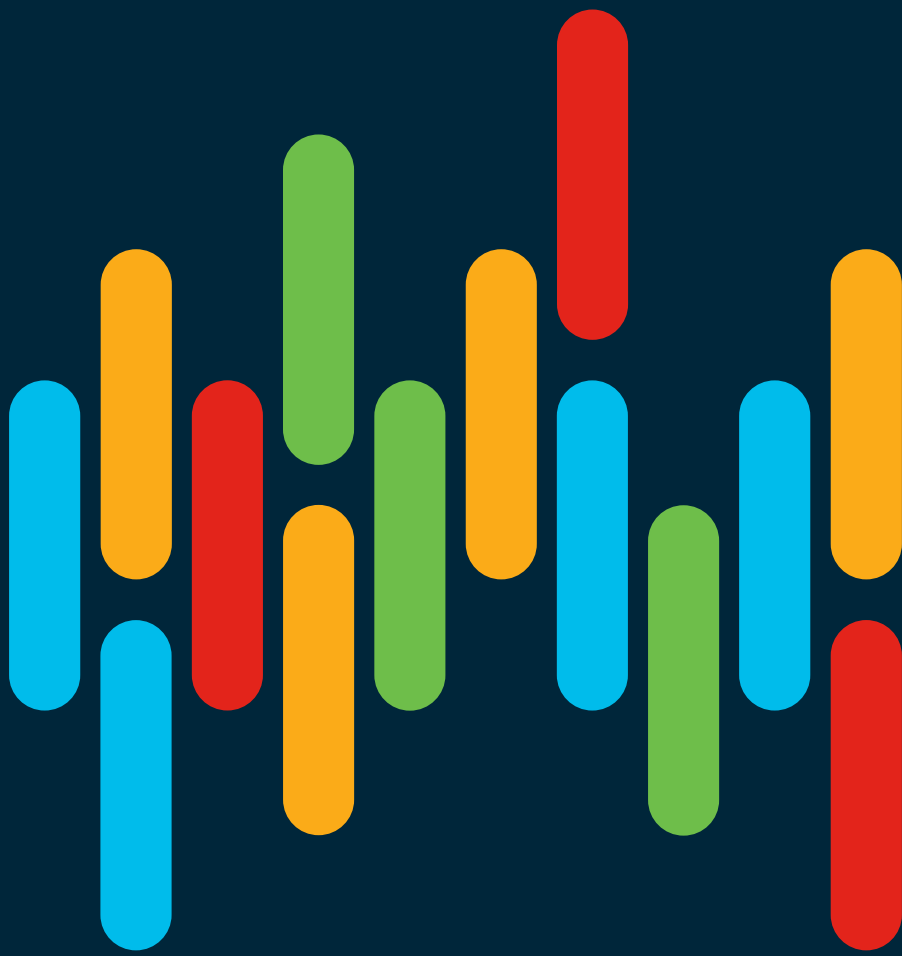
Continue your education

Demos in the Cisco campus

Walk-in self-paced labs

Meet the engineer 1:1 meetings

Related sessions



Thank you

Cisco live!

#CLUS

You make **possible**