
JUNIPER
NETWORKS
CONTAINERS AND MICROSERVICES WITH CONTRAIL
NXTWORK2017
JUNIPER CUSTOMER SUMMIT
Scott Sneddon
Sr. Director
Cloud and SDN
Sree Sarva
Sr. Director
Contrail Solutions
DP Ayyadevara
Director
Product Line Management

LEGAL DISCLAIMER

This statement of direction sets forth Juniper Networks' current intention and is subject to change at any time without notice. No purchases are contingent upon Juniper Networks delivering any feature or functionality depicted in this presentation.

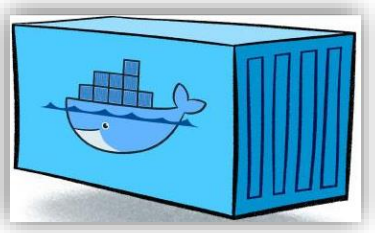
This presentation contains proprietary roadmap information and should not be discussed or shared without a signed non-disclosure agreement (NDA).

AGENDA

- ❑ **Background and Evolution of Technology and Ecosystem**
- ❑ **Fundamentals of Containers - what and why**
- ❑ **Microservices and Container Orchestration**
- ❑ **Contrail Networking for Containers**

CONTAINERS

THE RISE OF CONTAINERS & MICROSERVICES



Containers:

- Standardized frame for all services.
- Simplified a painful integration process in a heterogeneous infrastructure world.
- Docker - revolution in how developers build and deploy applications w/Containers.

APIs:

- The rapid adoption of APIs has created a standardized format for communications.

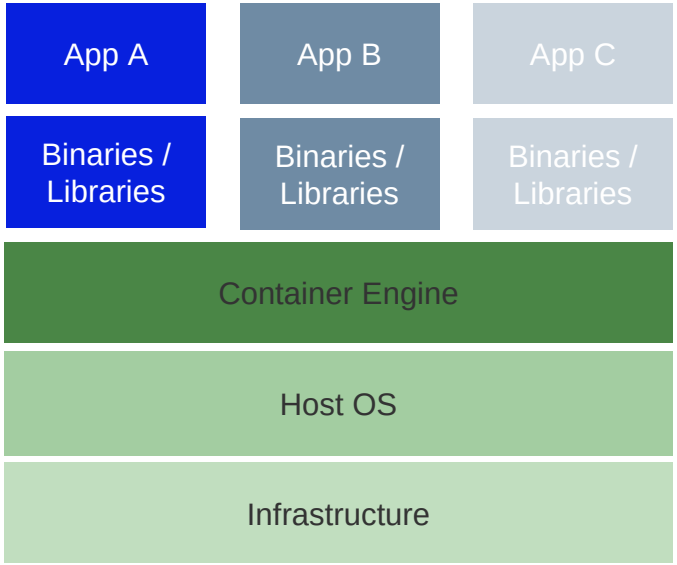
Scalable cloud infrastructure:

- Private or public, delivers the resources needed on demand to scale, and operate services effectively.
- Cloud is the new computer!

https://www.sequoiacap.com/article/build-us-microservices/

WHAT ARE CONTAINERS?

- Processes isolated from the host and (optionally) other containers
- Share the same underlying Kernel



- Virtual network interfaces / addresses (maybe host NAT'd)
- Files and optional (shared) mounts from the host filesystem

Control Groups (cgroups):
Virtualize by sharing and limiting access:

- CPU,
- Memory,
- Disk I/O,
- N/w I/O

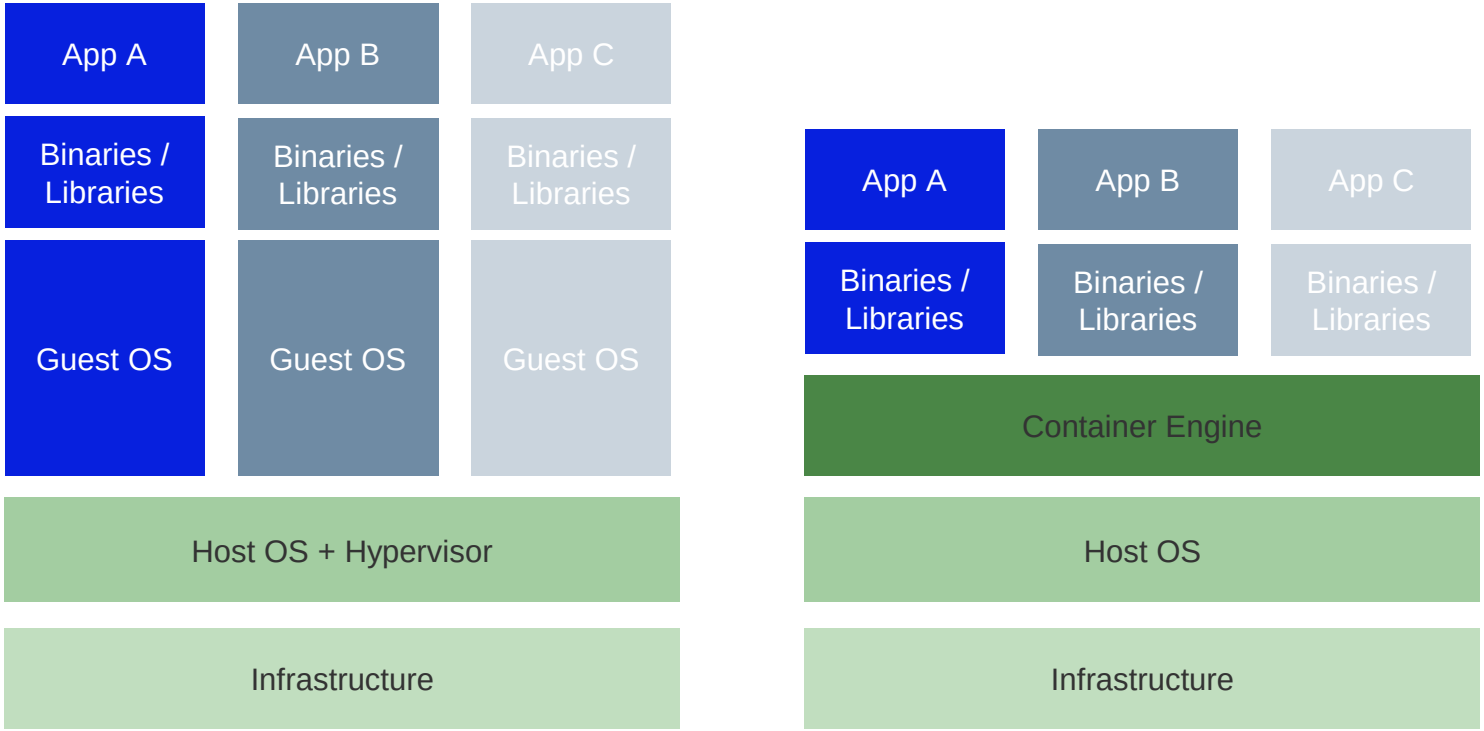
Namespaces:
Virtualize by isolating:

- User IDs
- Process IDs & tree
- Filesystem mounts
- Network interfaces

Security:

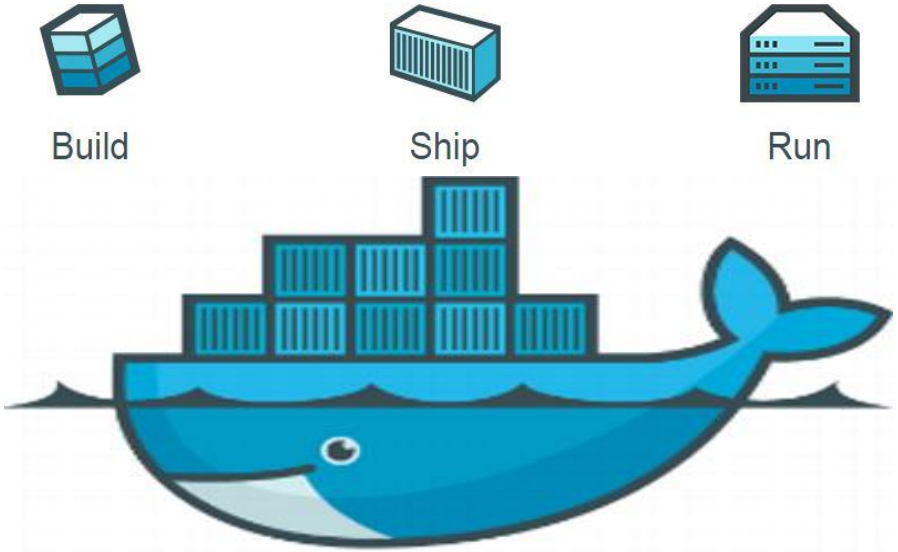
- **SELinux** policy and enforcement control over all resources
- **AppArmor** to restrict a program's abilities
- Linux capabilities etc.

VIRTUAL MACHINES VS CONTAINERS

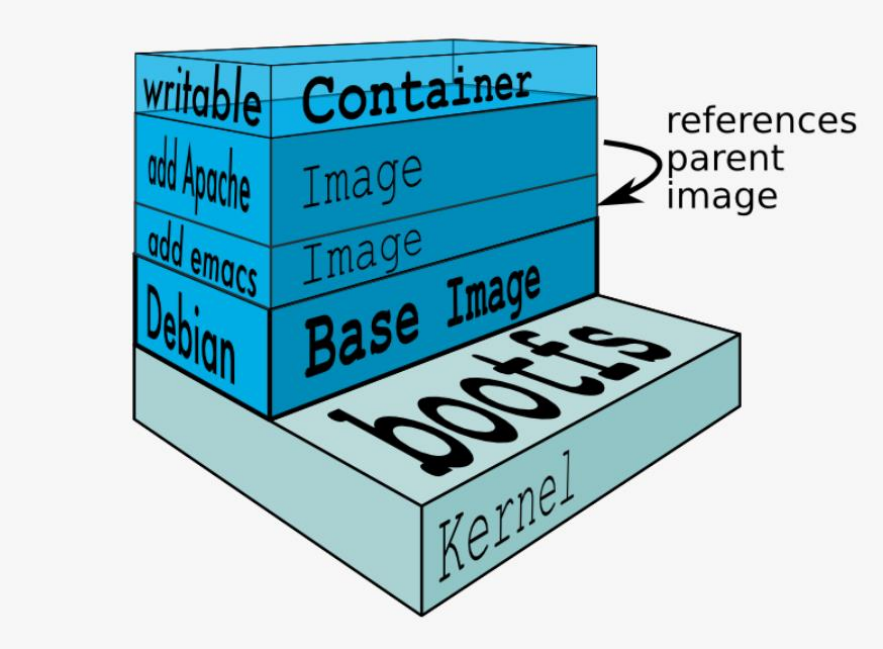


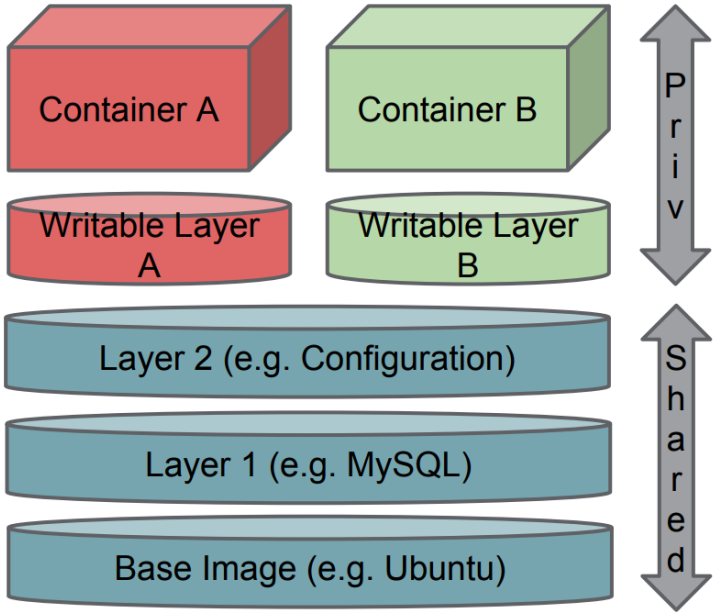
THE PROMISE OF DOCKER

Build Once, Run Anywhere

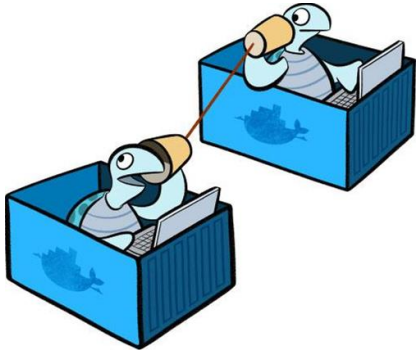


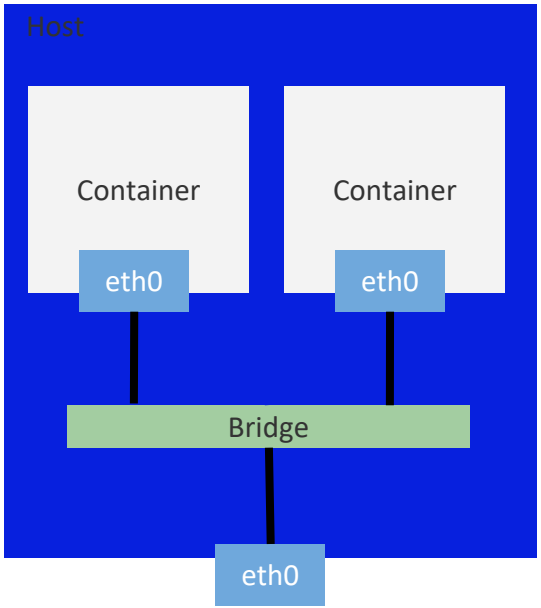
DOCKER LAYERS

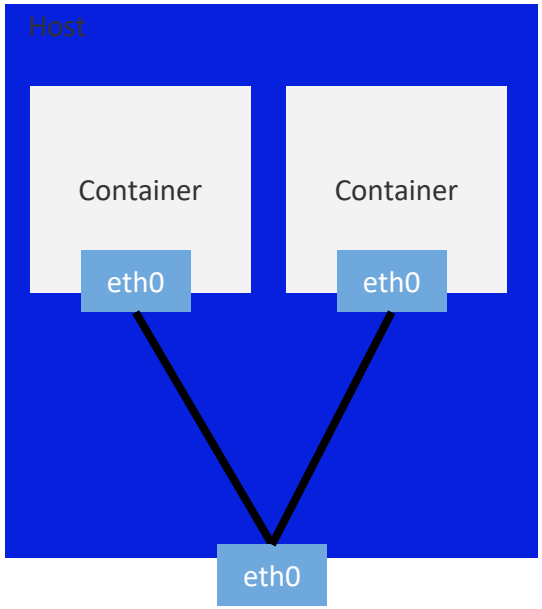




DOCKER NETWORKING



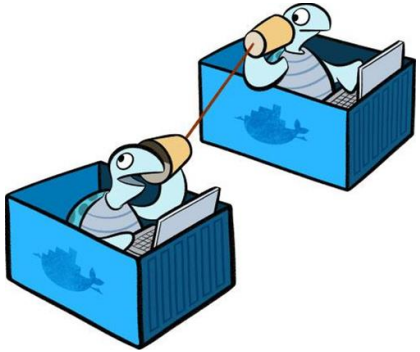




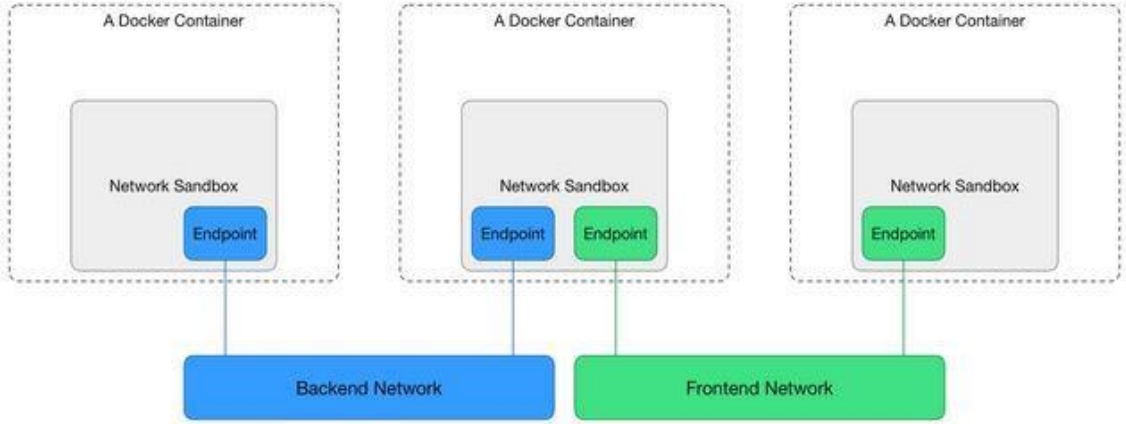
- Bridged (docker0)
- Bridged MacVLAN L2

- Host mode
- L3 mode

DOCKER NETWORKING



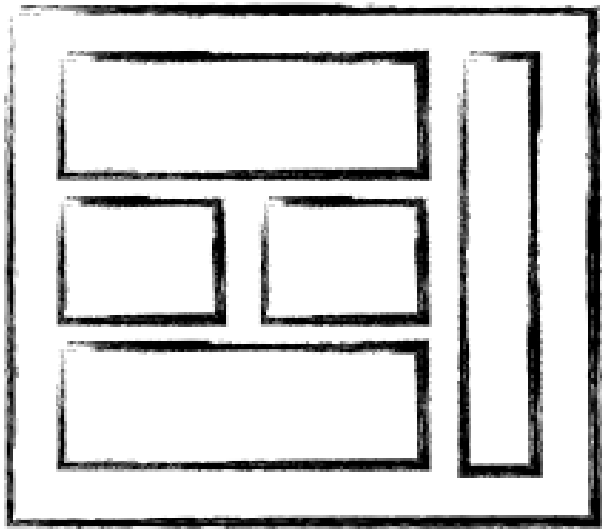
- **Docker provides:**
  - **Default networking**
  - **Mac-vlan**
  - **VXLAN Overlay**
  - **“multi-host” networking ;)**
- **Plugins**



https://blog.docker.com/2015/04/docker-networking-takes-a-step-in-the-right-direction-2/

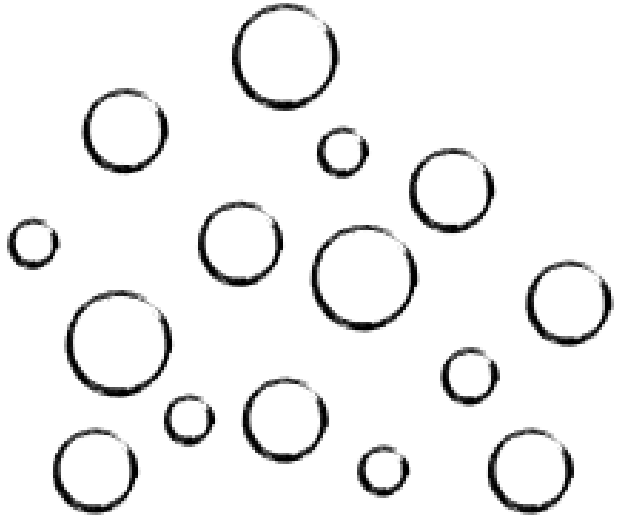

MICROSERVICES

MICROSERVICE CONCEPT



MONOLITHIC/LAYERED

(2006)

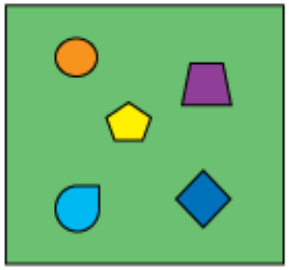


MICRO SERVICES

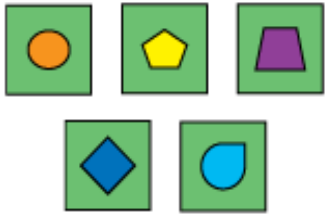
(2016)

MICROSERVICE CONCEPT

Monolithic app

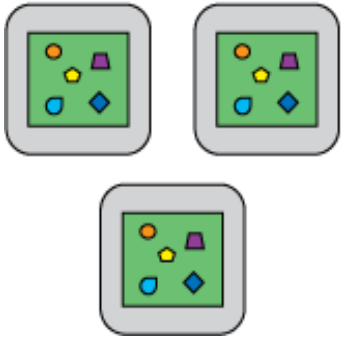


Micro services

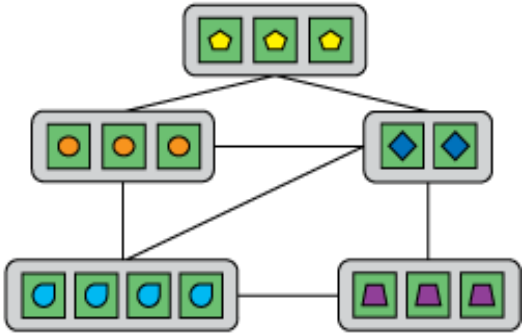


containers are the perfect vehicle for microservices

Scaling



Scaling



containers orchestration platforms are suited to handle the deployment and scaling of these microservices

CONTAINERS & MICROSERVICES ORCHESTRATION

KUBERNETES

http://www.publicartarchive.org/work/helmsman

The name ?

Greek for “pilot” or “helmsman of a ship”

The project?

Open Source project originally designed by Google (Project Borg) and donated to the CNCF

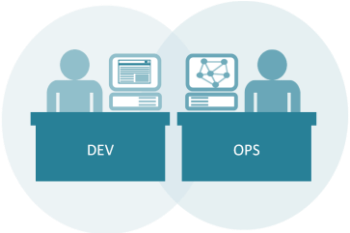
“Open-source platform for automating deployment, scaling, and operations of application containers across clusters of hosts, providing container-centric infrastructure”

CLOUD NATIVE COMPUTING FOUNDATION

Desired state / Intent-based system

“Tell Kubernetes what you want and it will take care of everything else”

WHAT IS KUBERNETES ?



Application Focus

- Deploy / scale / rolling updates / control resources

Portable

- Run everywhere (public/private cloud, bare metal, VMs, laptop)

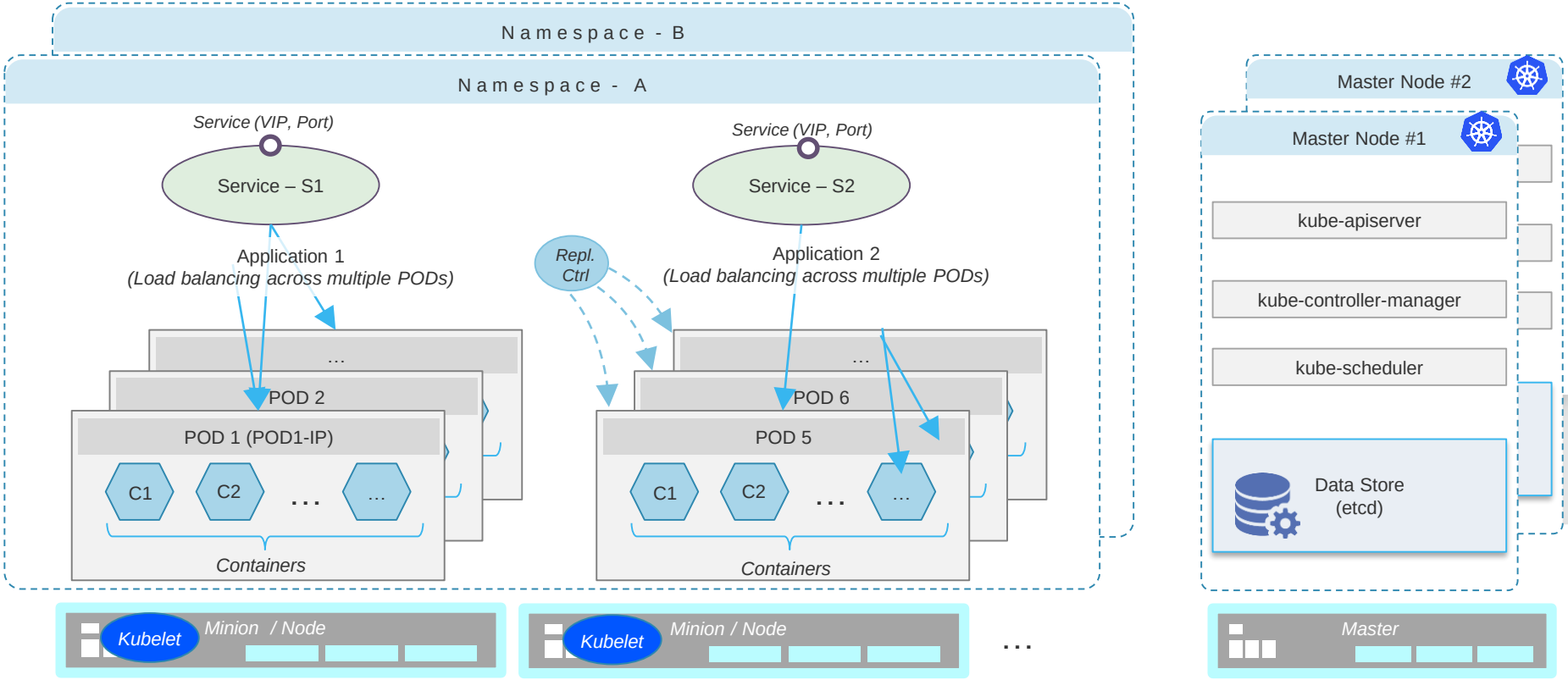
Self-Healing

- Auto-restart, auto-rescheduling, auto-scaling, auto-replication

Bridge the Dev/Ops gap

- Taking advantage of application containerization and micro services
- CI/CD environments
- Mechanisms to tie the infrastructure and the applications running on it

KUBERNETES ARCHITECTURE



OPENSHIFT

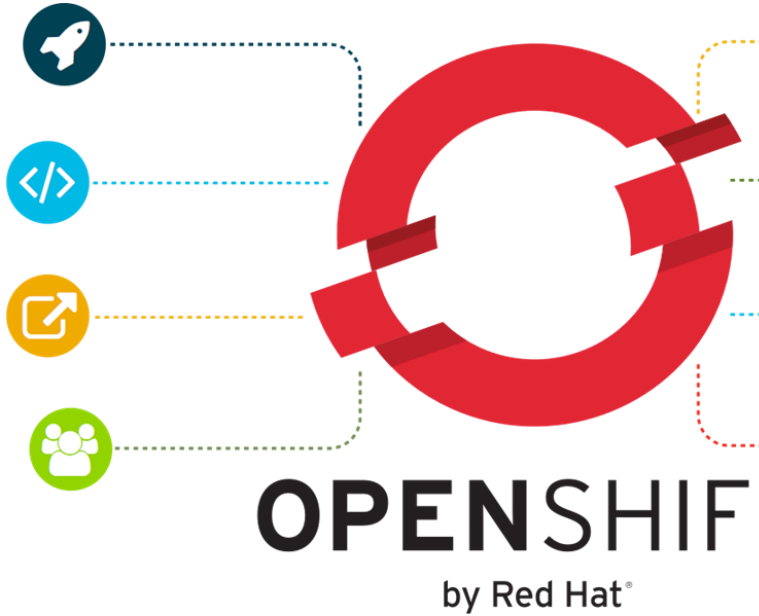
Red Hat's Container Application Platform (PaaS)

Self Service
- Templates
- Web Console

Multi-language

Automation
- Build
- Deploy

Collaboration
- DevOps



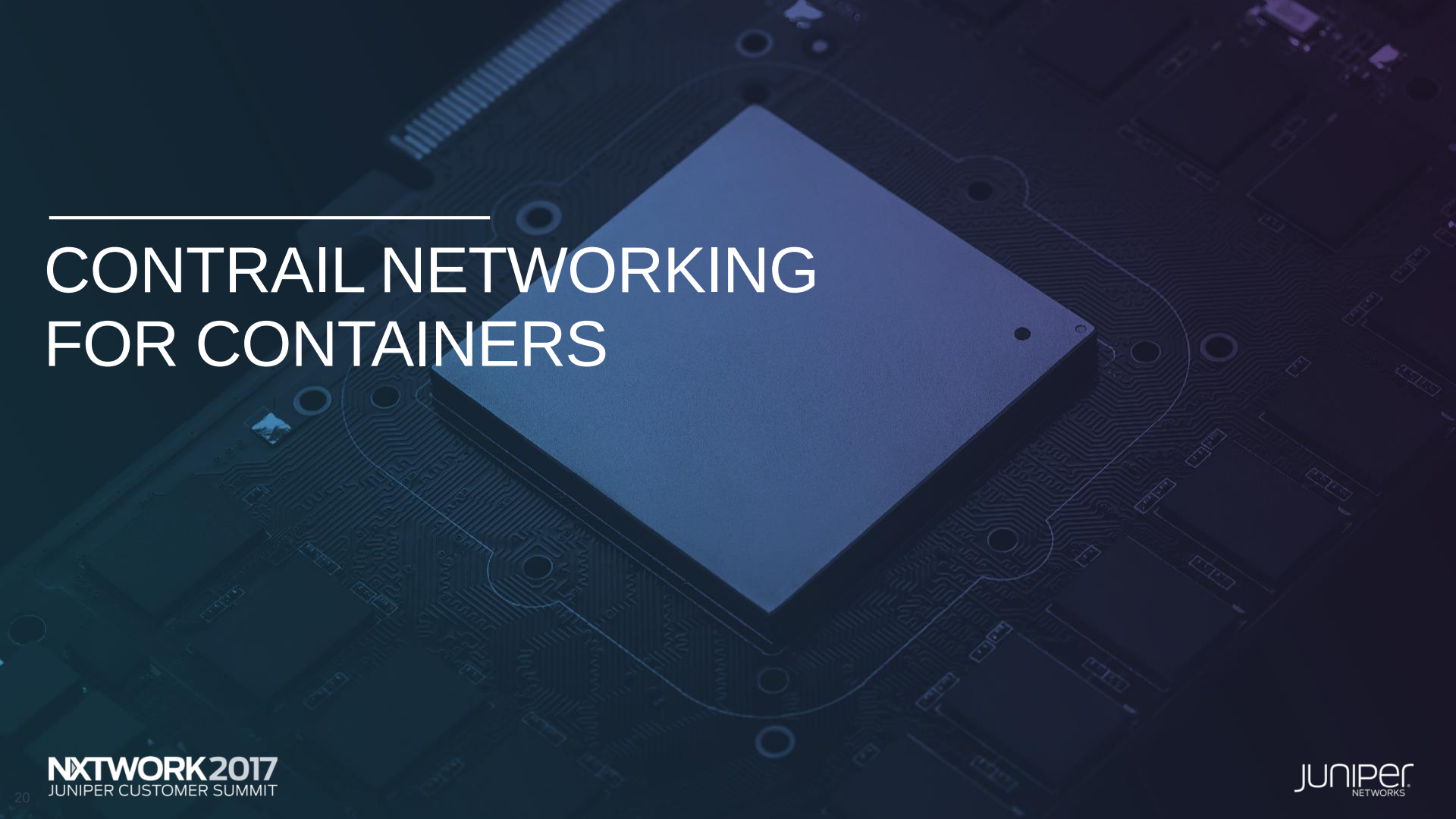
Security
- NameSpaces
- RBAC

Scaleable
- Integrated Loadbalancer

Open Source

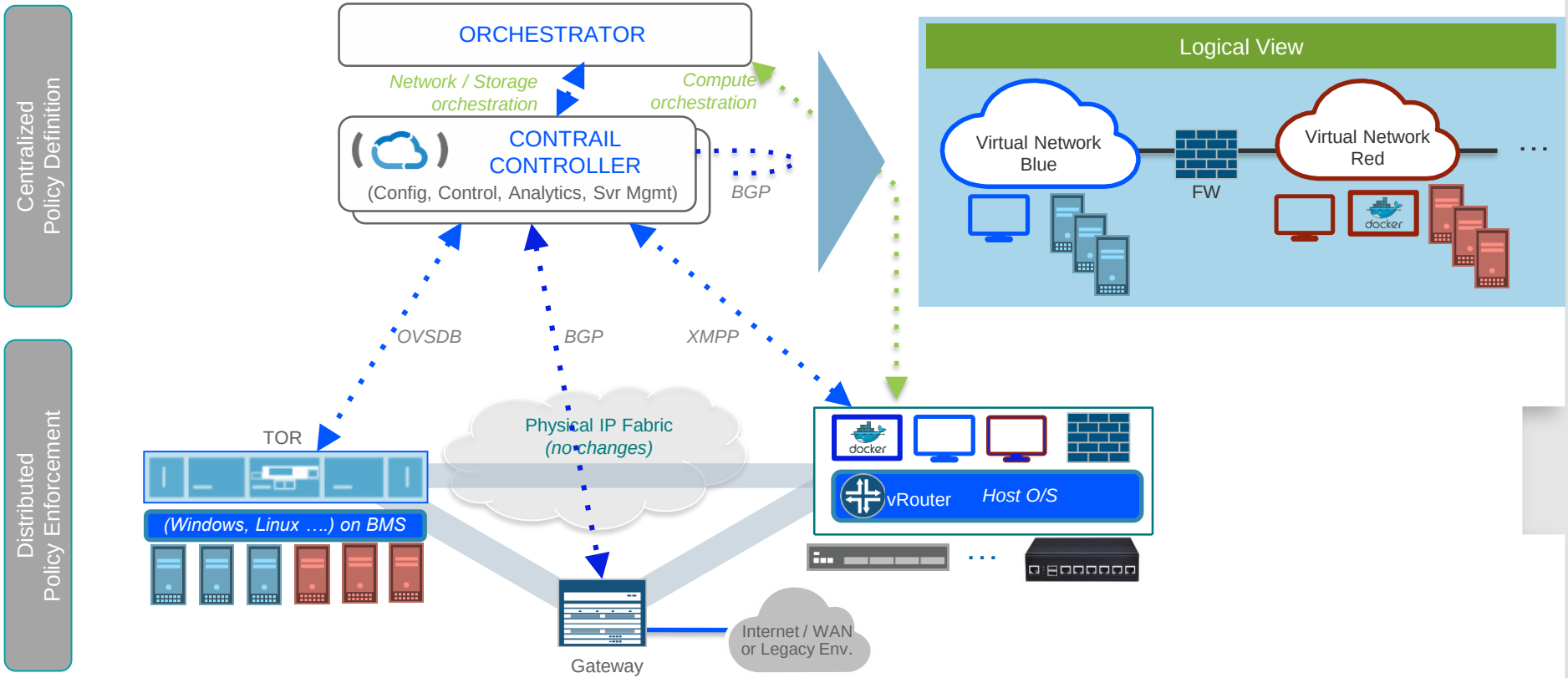
Enterprise Grade
- Authentication
- Web Console
- Central Logging

source: www.redhat.com



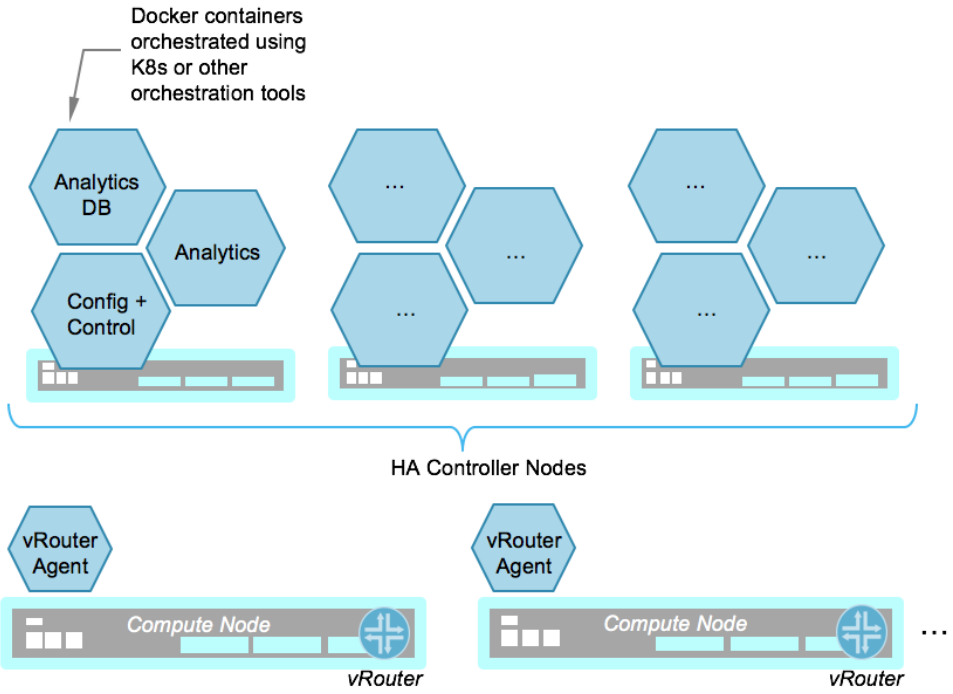
CONTRAIL NETWORKING FOR CONTAINERS

CONTRAIL ARCHITECTURE



CONTAINERIZED CONTRAIL

Containerizing Contrail Control Plane – for easier manageability



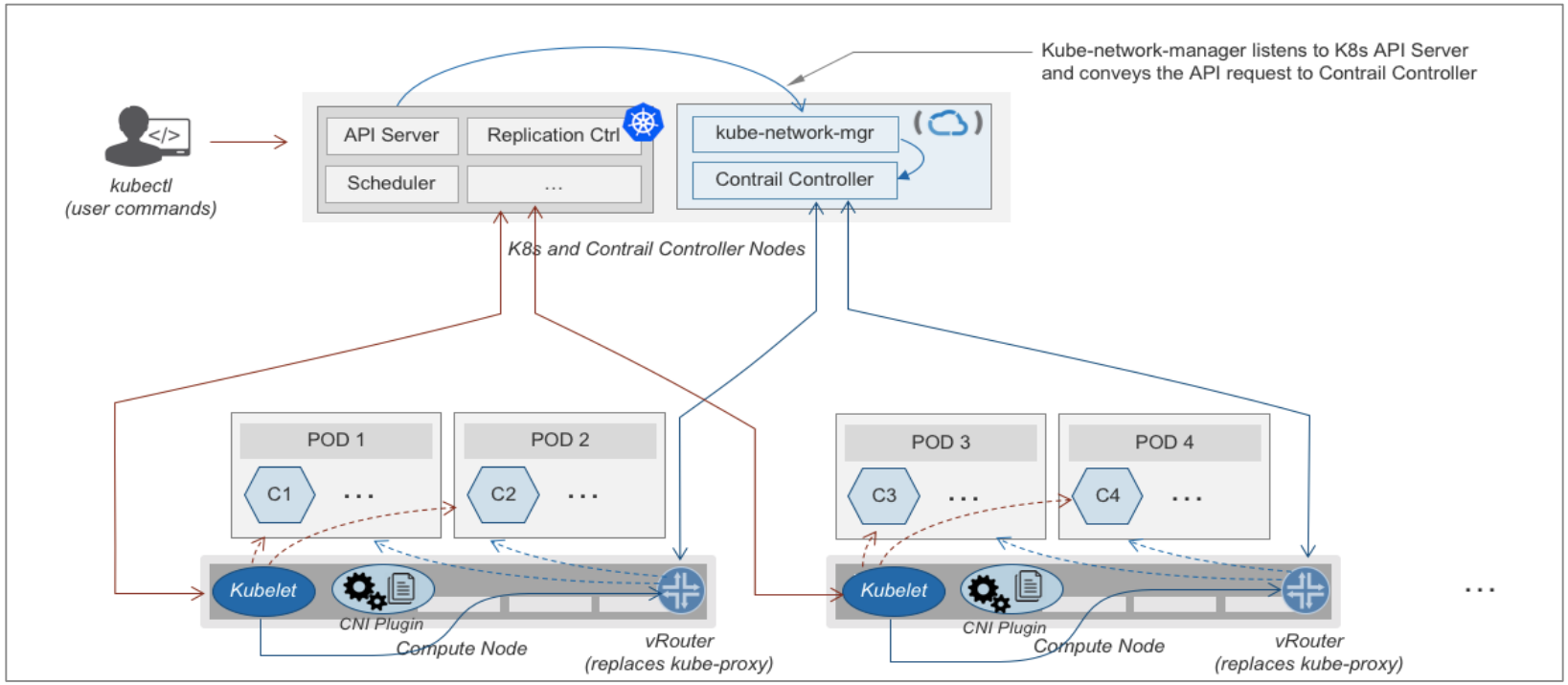
SALIENT ASPECTS

- Multiple personalities of containers:
  - 3 controller container – (Controller, Analytics, Analytics DB) each representing a node
  - LB to enable HA (based on HAProxy) will be provided as container not a mandatory item
  - vRouter Agent on containers
- Containers are deployed using either Contrail Server Manager / K8s / Helm Charts
- Each of the nodes can independently scale (3 x)
- Can be deployed on Bare Metal or VMs
- No change in the role / functionality of the Control / config / analytics nodes

BENEFITS

- LCM is simplified [All dependencies within the container (easy bring up)]
- Accelerate Contrail provisioning
- Integration with 3rd party provisioning tools simplified

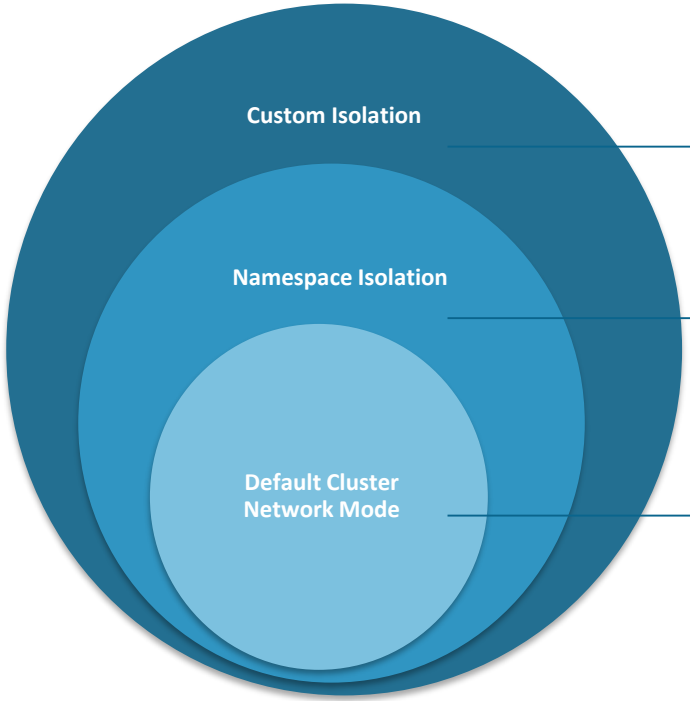
CONTRAIL ARCHITECTURE WITH KUBERNETES



CONTRAIL WITH KUBERNETES

CONTRAIL VALUE PROP

INCREASING LEVELS OF ISOLATION



- Define Networks and Security Policies in Contrail
- Use Network/Security Policy per namespace and/or pods using annotations

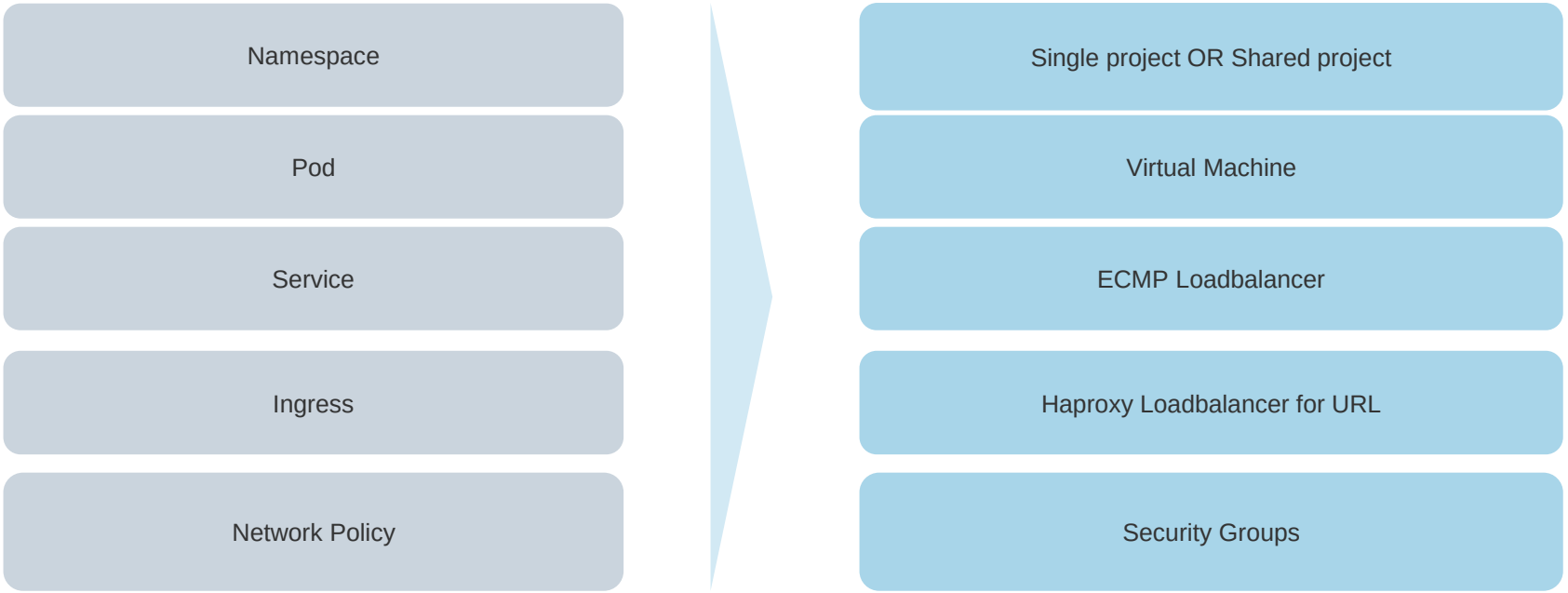
- Virtual Network per namespace
- solation per namespace

- All Pods and Services are reachable

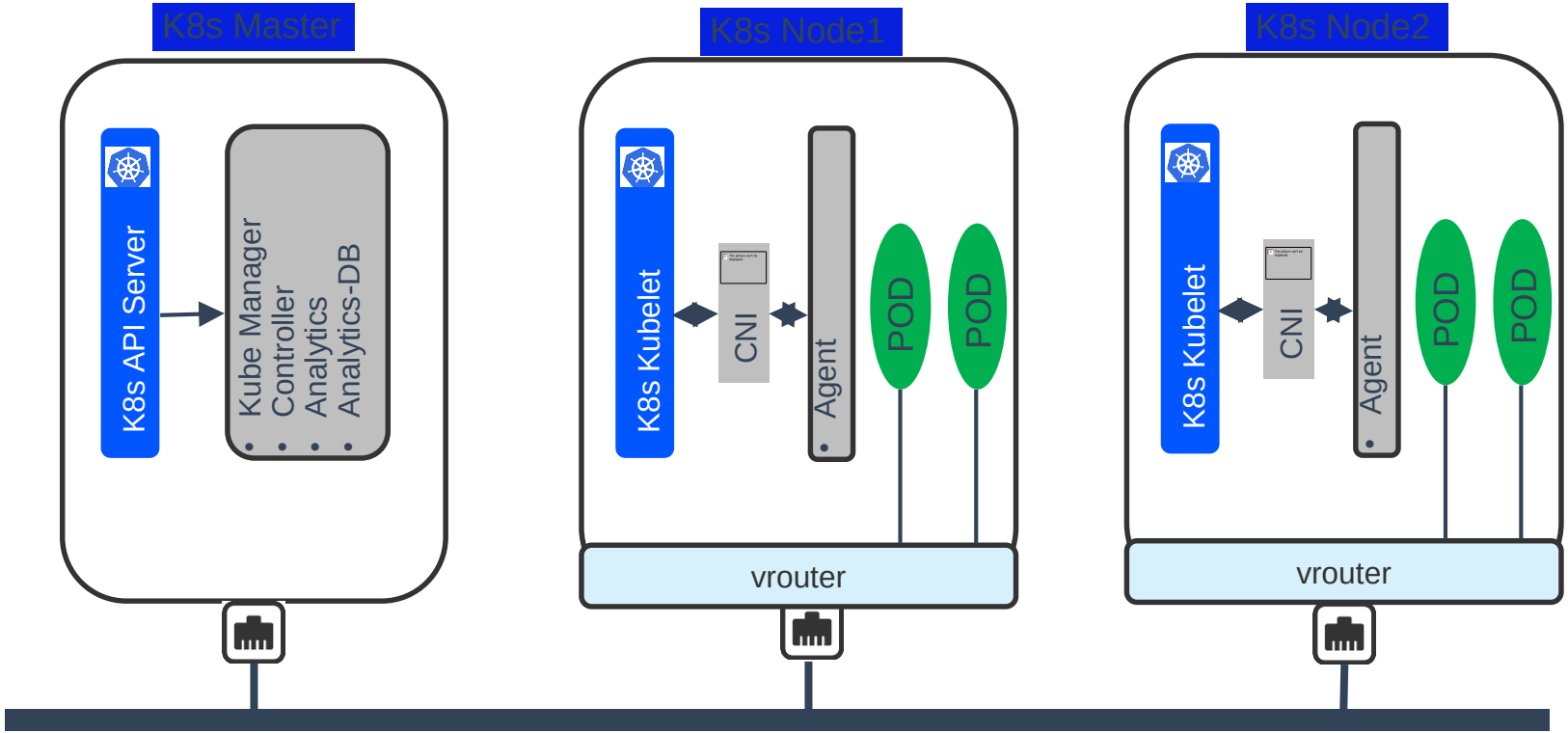
What do the users / app team get ?

- Distributed LB
- Security / Isolation / multi-tenancy
- External / Public access from within cluster (SNAT) and vice versa (using floating IP)
- Exposing cluster to Enterprise network outside the cluster

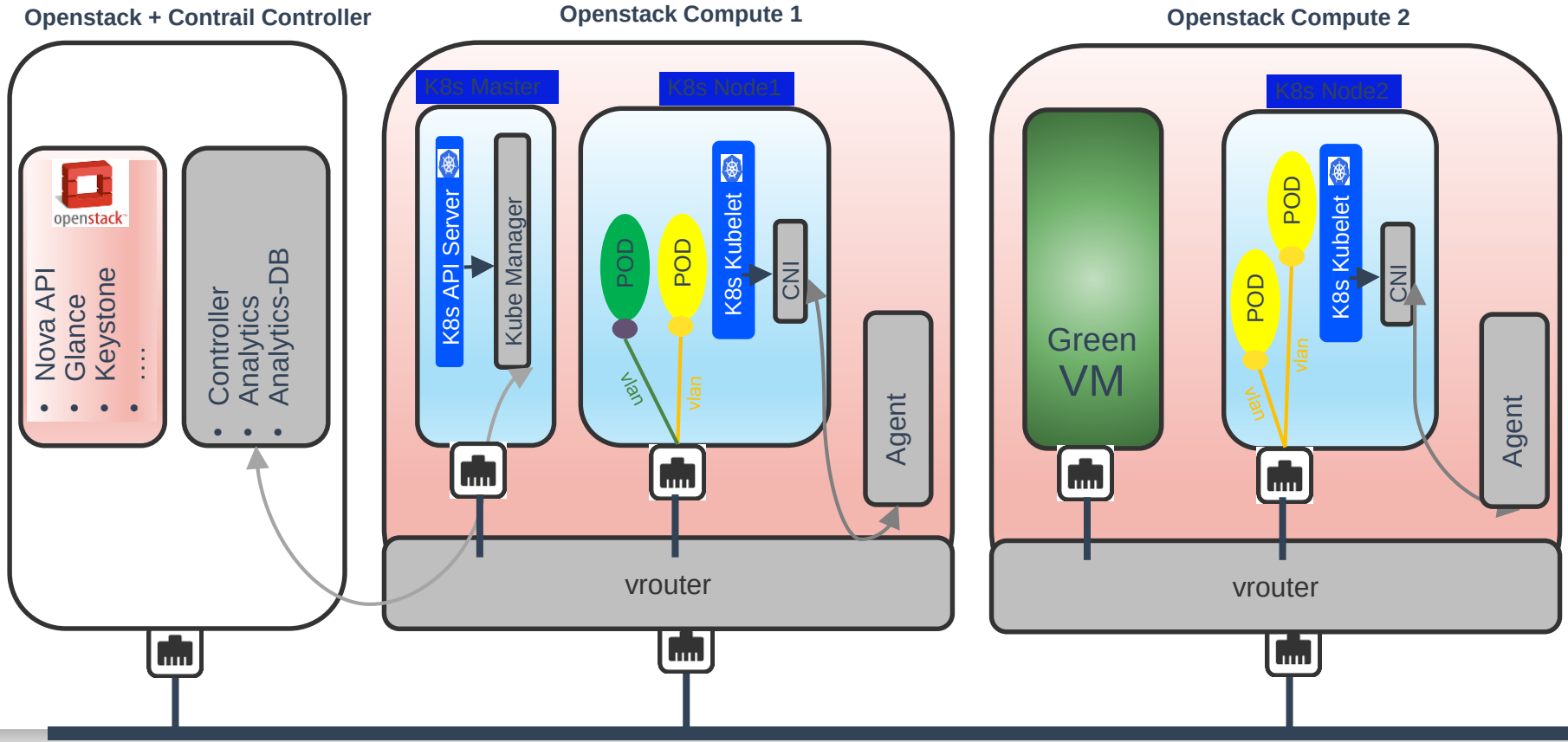
KUBERNETES TO CONTRAIL OBJECT MAPPING



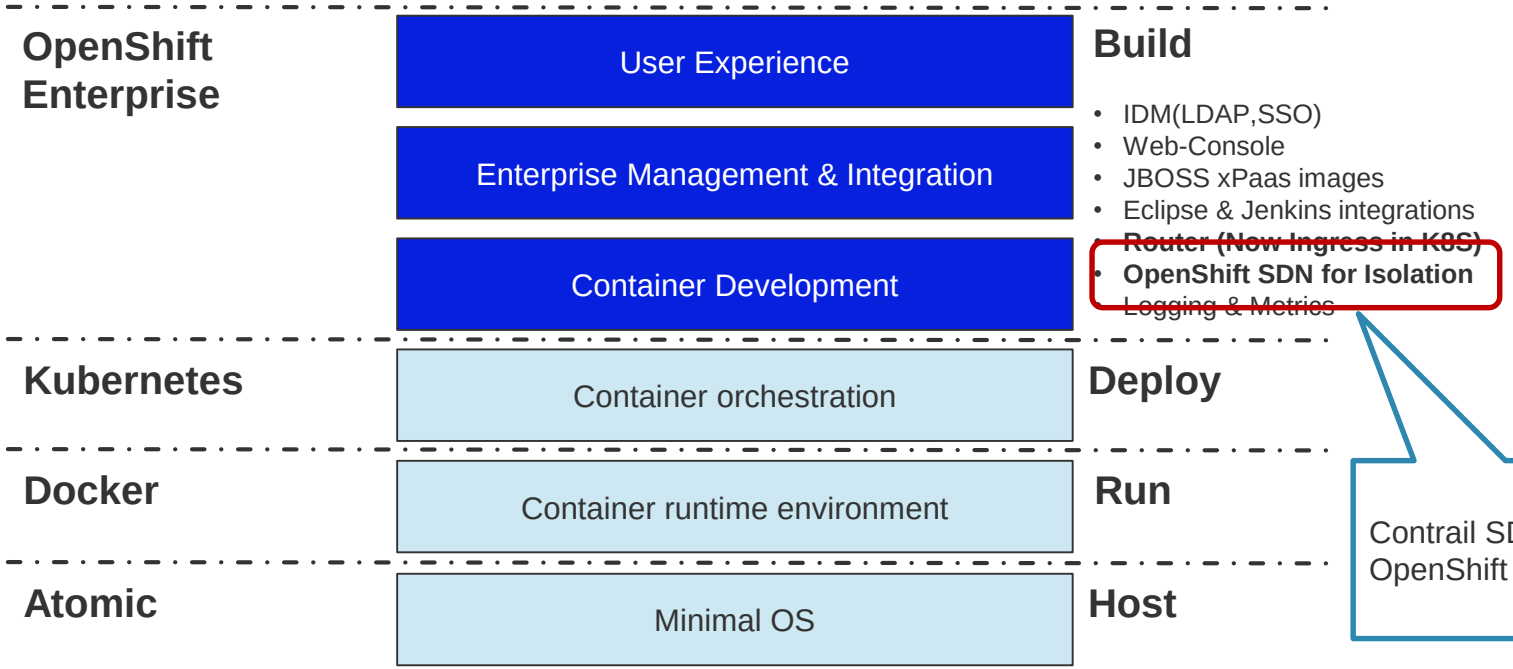
CONTRAIL WITH KUBERNETES ON BAREMETAL

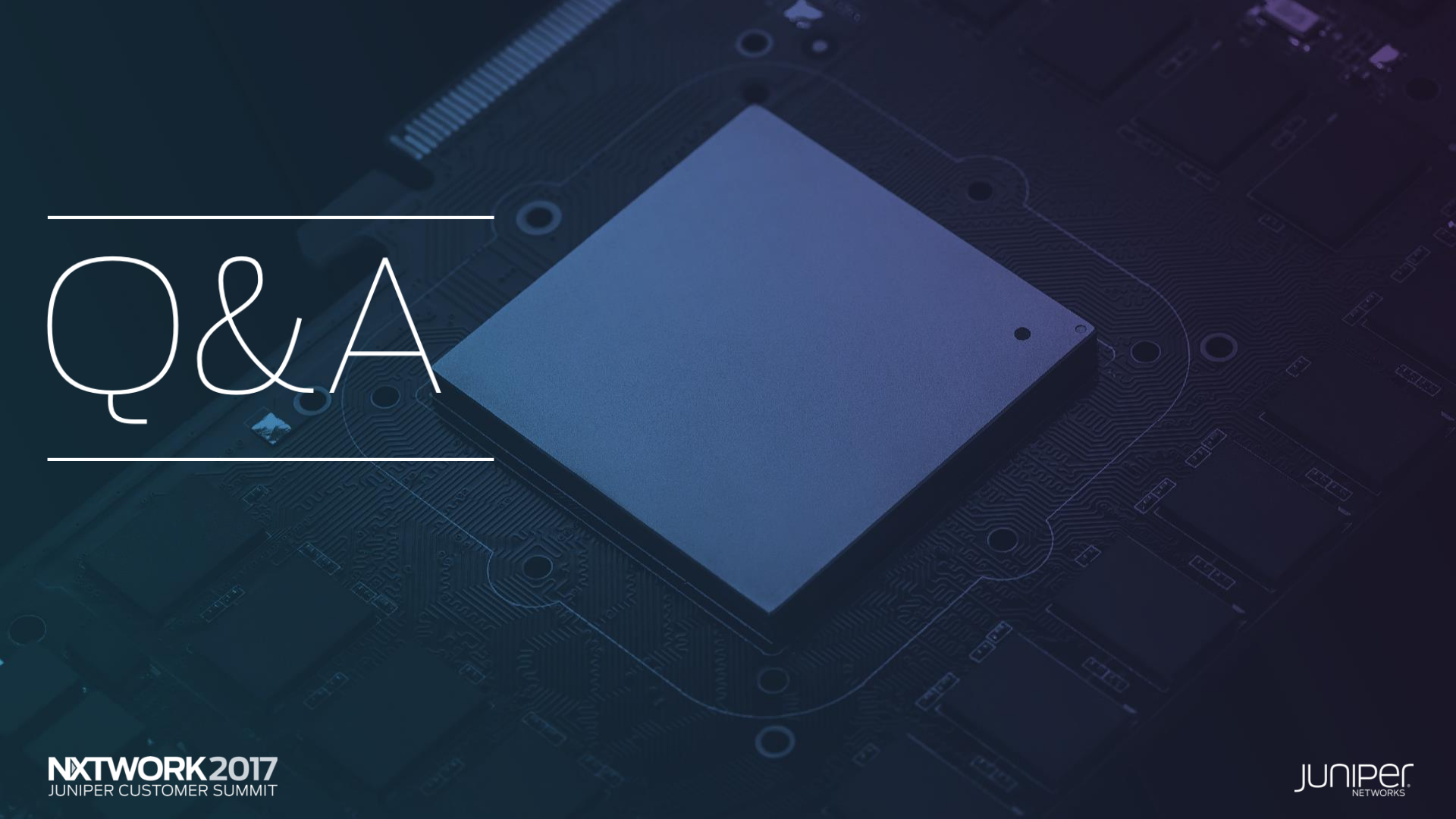


CONTRAIL WITH KUBERNETES ON OPENSTACK



CONTRAIL WITH OPENSHIFT




Q&A