RED HAT FORUM 2018 ZURICH

redhat.

OPENSHIFT OPTIMIZED BY INTEGRATING SOFTWARE DEFINED INFRASTRUCTURE

RUBEN KNAUS	CLOUD ENGINEER
DANIEL KASPER	CLOUD ENGINEER
SAMUEL **ZEHNDER**	SENIOR CLOUD ENGINEER

NETCLOUD
ICT PROFESSIONALS

Über Netcloud

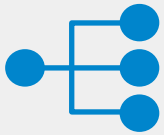
NETCLOUD
ICT PROFESSIONALS

Niederlassungen	Winterthur, Basel, Bern
Service Operations-Center	Winterthur, Bern
Gründungsjahr	1998
Mitarbeiter	164
Red Hat	CCSP Advanced Partner
Cisco Partner Level	Gold
NetApp Partner Level	Platinum
AVI	Premium Partner
Besitzverhältnisse	Zu 100% in Besitz des Managements

Cisco ACI CNI für OpenShift

Enhanced OpenShift Networking

Unified networking: Containers, VMs, and bare-metal



Hardware-accelerated: Integrated load balancing

Flexible policy: Native platform policy API and ACI policies

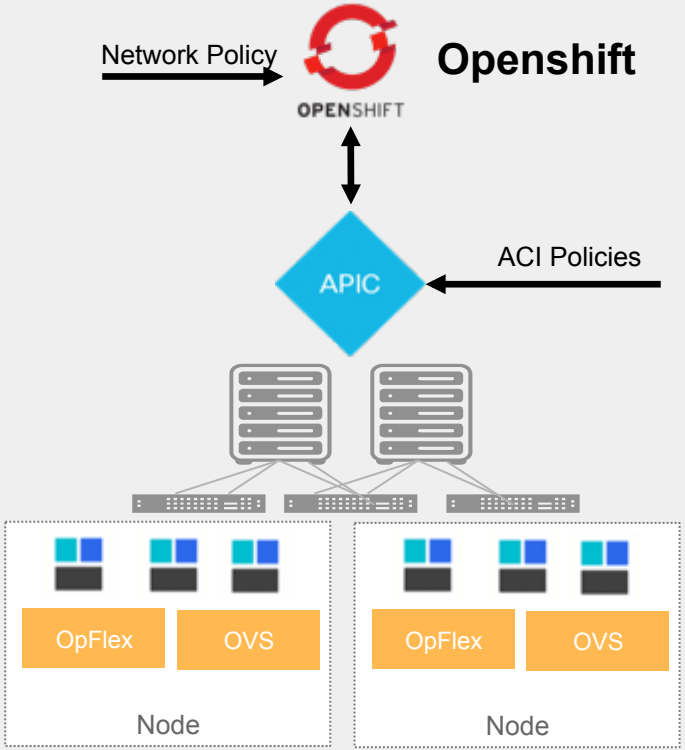
Visibility: Live statistics in APIC per container and health metrics

Enhanced Multitenancy and unified networking for containers, VMs, bare metal

Fast, easy, secure and scalable networing for your Application Container Platform

redhat.

Solution Overview



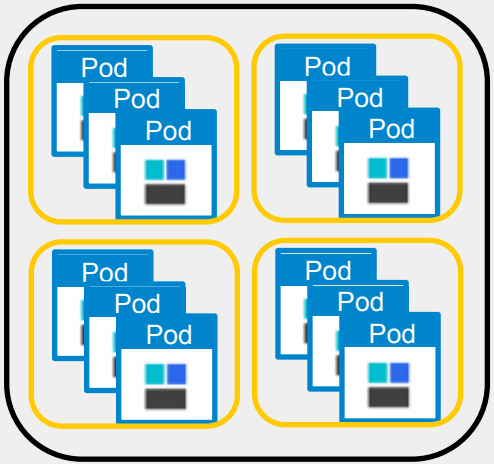
- Network policies of Kubernetes supported using standard upstream format but enforced through OpFlex / OVS using APIC Host Protection Profiles
- Openshift apps can be moved without modification to/from ACI and non-ACI environments
- Embedded fabric and virtual switch load balancing
  - PBR in fabric for external service load balancing
  - OVS used for internal service load balancing
- VMM Domain for Openshift
  - Stats per namespace, deployment, service, pod
  - Physical to container correlation

EPG

Network Policy

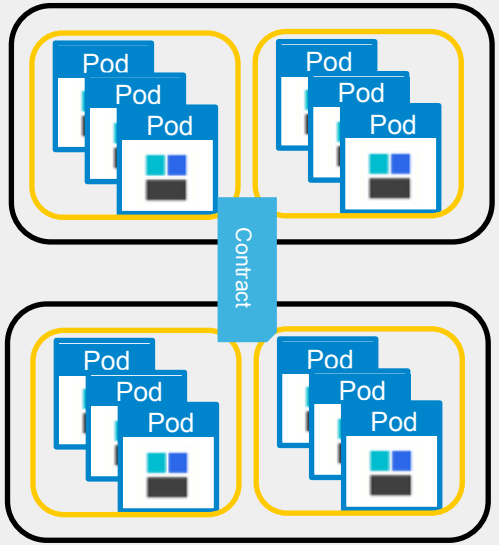
Flexible Definition of EPG Boundaries

Cluster Isolation



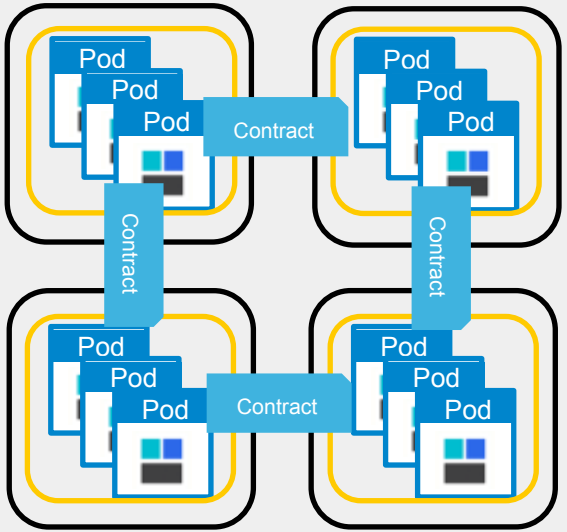
- Single EPG for entire cluster, default behavior
- No need for any internal contracts

Namespace Isolation



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

Deployment Isolation



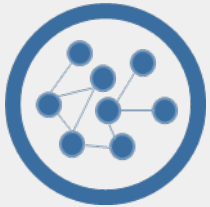
- Each deployment mapped to an EPG
- Contracts tightly control service traffic

redhat.

AVI Elastic Service Mesh für OpenShift

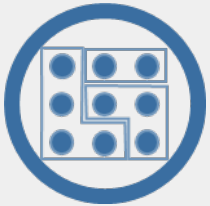
Was ist AVI ESM

Application
Perf. Monitoring



Application
Maps

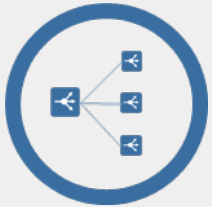
Service
Discovery



Security /
Microsegmentation

AVI Networks

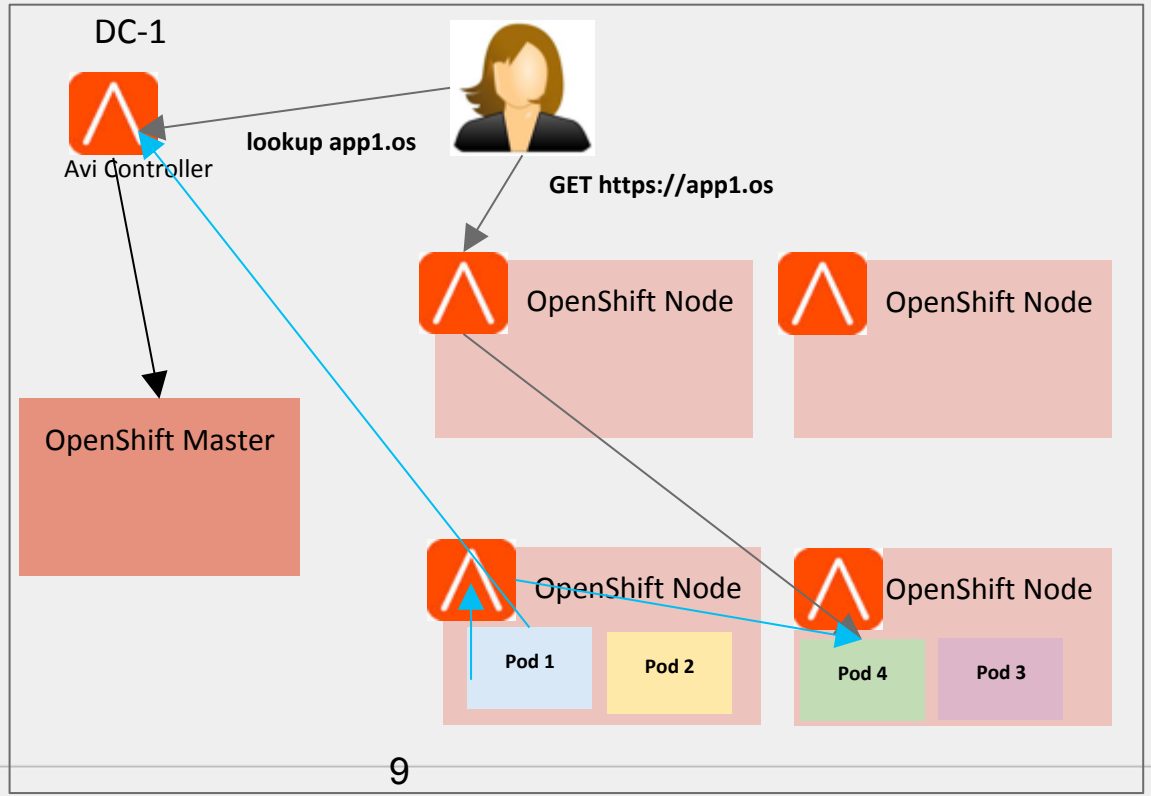
Elastic Service Mesh



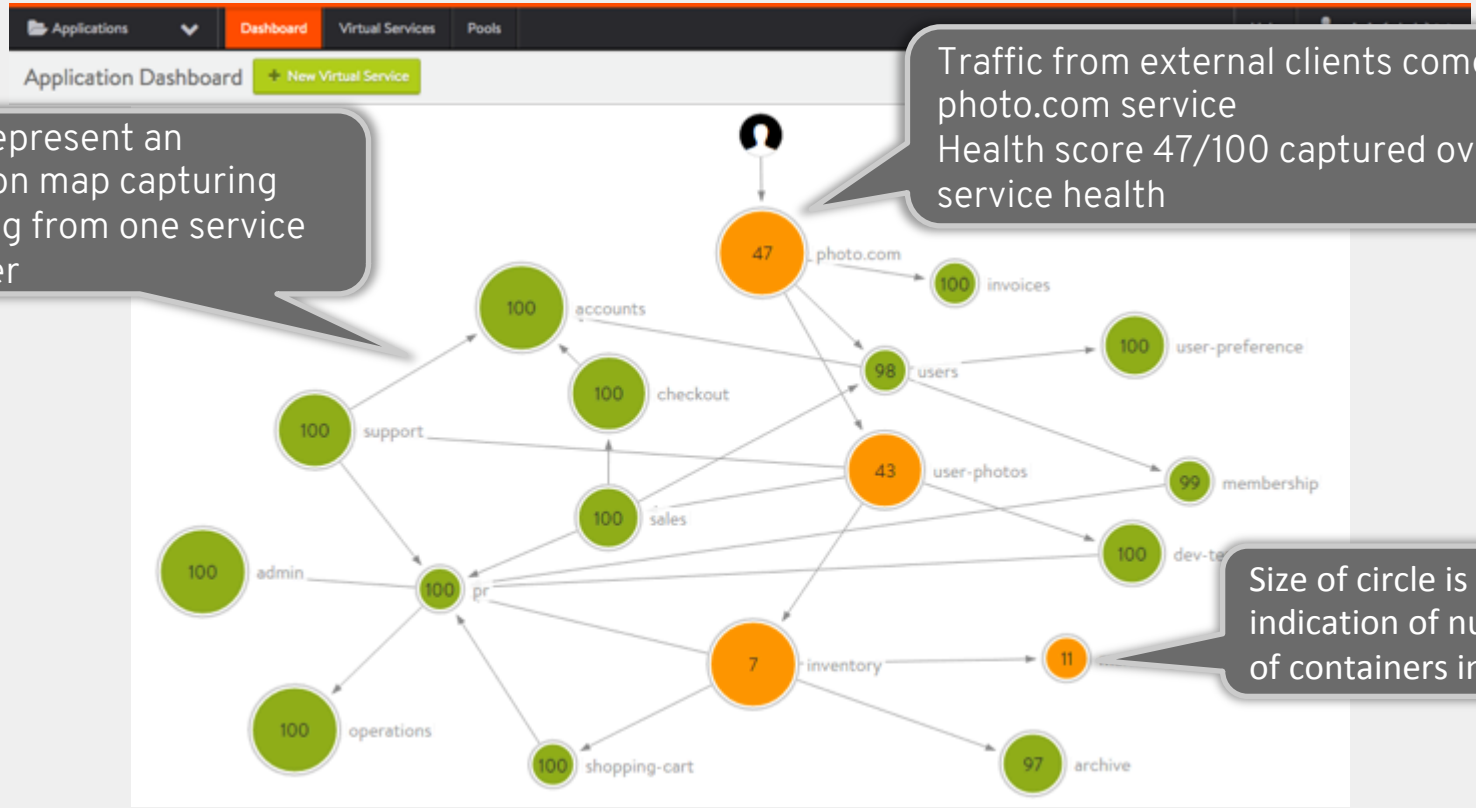
Global / Local
Load Balancing

Traffic Flow

North/ South Traffic
East/ West Traffic



Service Map



Benefit of AVI ESM

- Load Balancing
  - L4-7 Load Balancing with Auto-Scale
  - L4-7 Traffic Management
- Microsegmentation/firewalling for containers
- Integrated IPAM + DNS - service discovery
- Fully automated proxy + service creation
- App Upgrades and Blue/Green App Deployments
- Works in any enviroment
  - DC, AWS, Azure, GCP, etc.
- Monitoring
  - Real-time visibility into application health
  - Application End-to-End latency tracking
- Management
  - Centralized Monitoring und managemnet

Kubernetes Persistent Volume Framework

Persistent Volumes (PVs)

- Storage which has been introduced to Kubernetes by an administrator
- Configured for backing storage device like NFS, iSCSI, Cinder, AWS EBS, GCE, Azure, …
- Abstracts the physical storage volume into an allocatable unit for applications
- Includes connection information for the storage volume

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv0003
spec:
  capacity:
    storage: 5Gi
  accessModes:
    - ReadWriteMany
  nfs:
    path: /tmp
    server: 172.17.0.2
```

Persistent Volumes Claims (PVCs)

- Created by a user to request storage
- Specifies desired capacity and access mode, along with labels to aid with selection
- Kubernetes assigns a PV to meet the requirements requested in the PVC

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: rhforum2018
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 5Gi
```

Storage Classes

- Describes a storage offering and associates a provisioner
- Parameters are used to provide additional information to the provisioner
- Parameters are opaque to Kubernetes
- PVCs specify storage classes, storage classes specify provisioners, and provisioners map storage classes to PVs
- Storage Classes can also be used with statically provisioned PVs

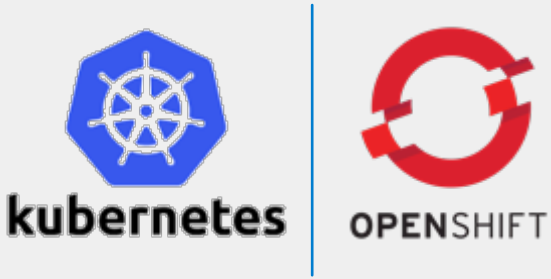
```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: bronze
provisioner: netapp.io/trident
parameters:
  backendType: "ontap-nas"
  mediaType: "hdd"
```

Persistent Volumes with NetApp Trident

Trident

- A dynamic and automated storage provisioner for Kubernetes and Red Hat OpenShift
- Supports ONTAP, SolidFire, and E-Series
- Abstracts back ends into pools of capabilities and retains the ability to differentiate storage
- IOPS, compression, disk type, etc. all able to be specified
- Maps storage requests to storage pools, each backend can contain one or more storage pools

[Download now on github]

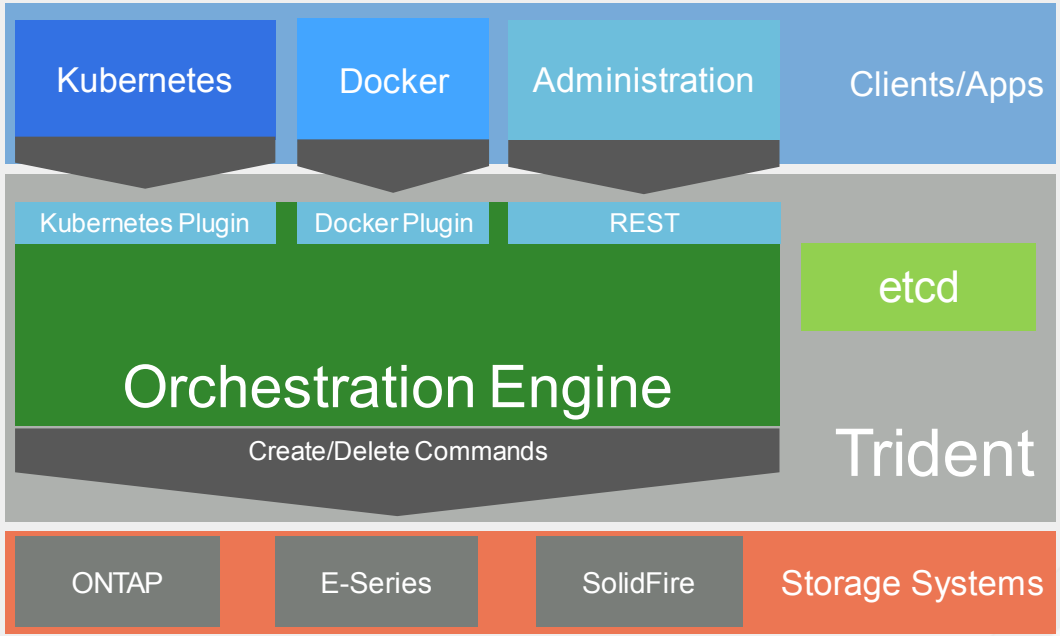


Trident Architecture

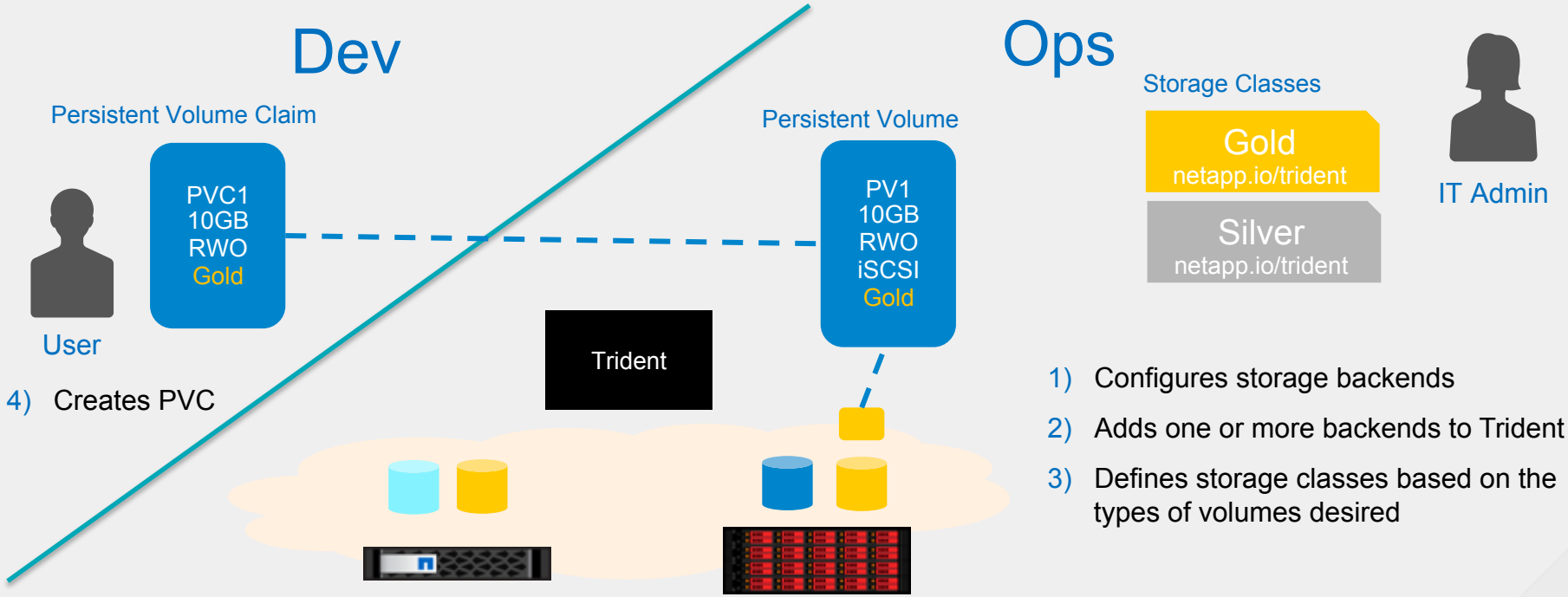
Clients integrate with Trident

Orchestration Engine decides what and where to create ressources

Backend Systems, ontap, e-series and SolidFire Support



Storage Provisioning



Cool Features I

CloneFromPVC

Create PVC from Clones

- Ability to clone existing volumes by setting a custom annotation in a newly created PVC
- Existing PV will be cloned in backend and presented as a new PVC to enduser

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: prod-clone
  annotations:
    trident.netapp.io/cloneFromPVC: prod
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: gold
```

Cool Features continued…

SnapshotDir, Encryption

Access NetApp Snapshots

- Admin can allow access to .snapshot directory where NetApp Snapshots are available

Encryption

- Admin can enable NetApp Volume Encryption (NVE)

Rate the Session in the Event App

RED HAT FORUM 2018

NEXT PRESENTATION

16:30

END NOTE & COCKTAIL RECEPTION