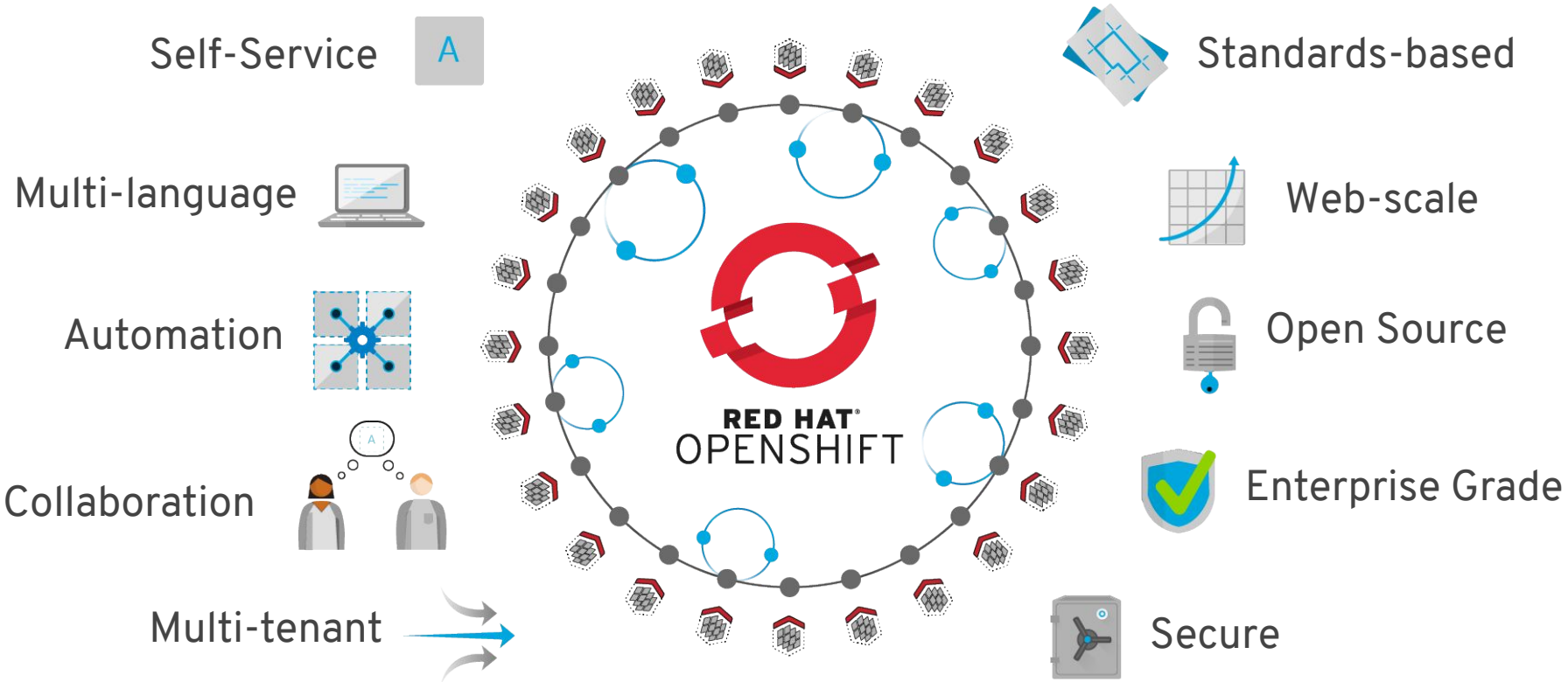
OpenShift 4.x Architecture Workshop

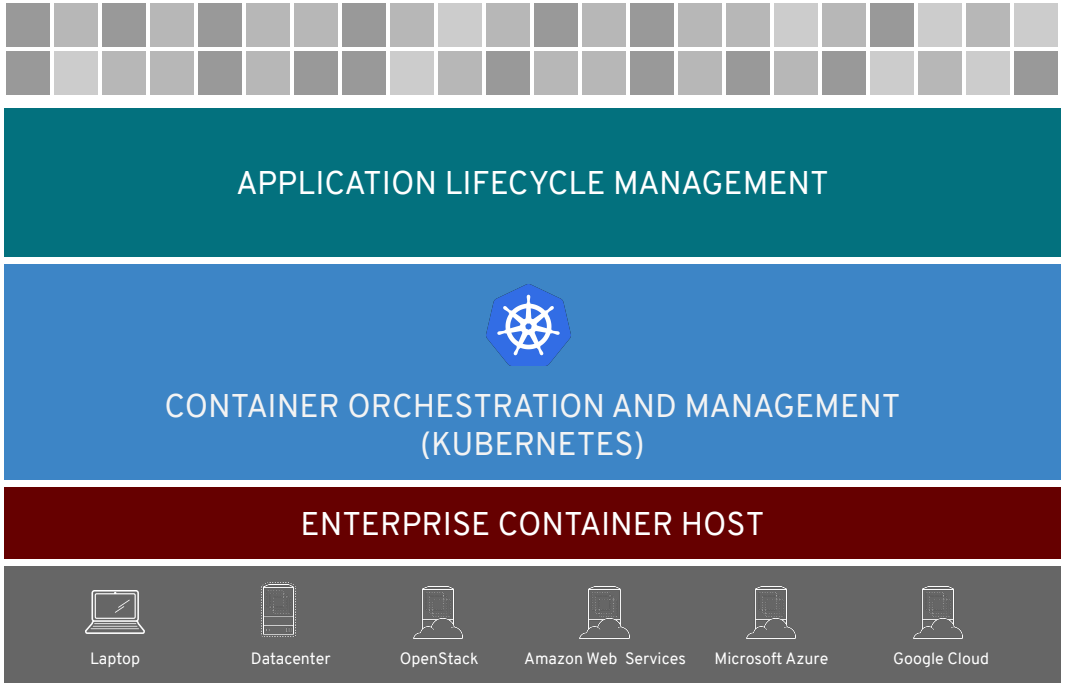
OpenShift Container Platform (OCP)
Advanced Architecture

July 2019

Red Hat


Self-Service
A
Multi-language
Automation
Collaboration
Multi-tenant
RED HAT
OPENSHIFT
Standards-based
Web-scale
Open Source
Enterprise Grade
Secure

OPENSHIFT CONTAINER PLATFORM



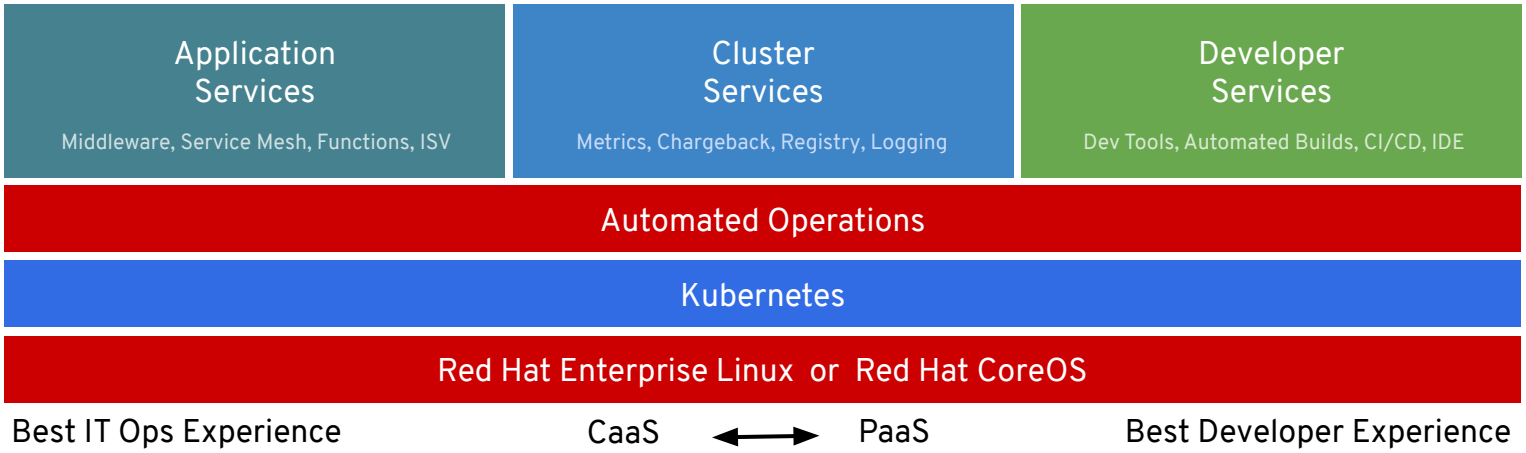
ANY
CONTAINER

RED HAT®
OPENSHIFT

ANY
INFRASTRUCTURE

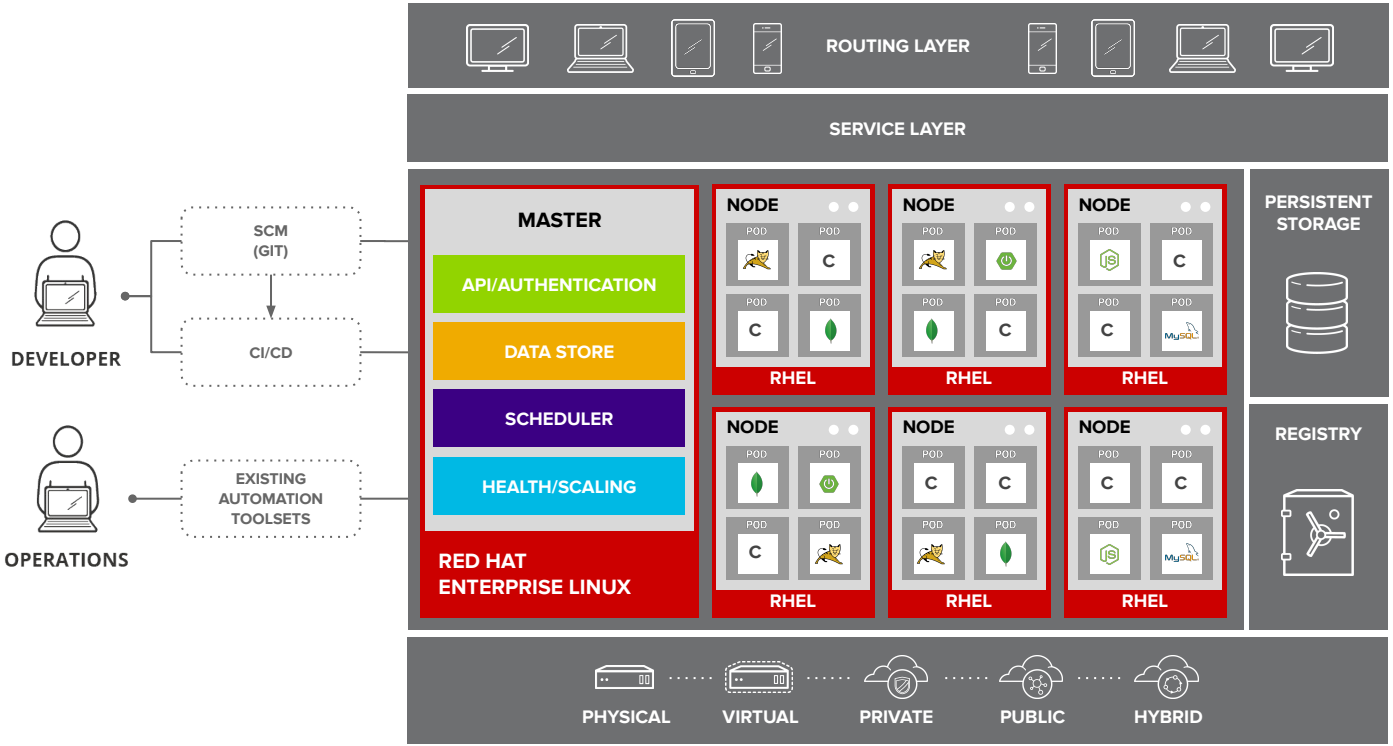
Red Hat

OPENSHIFT CONTAINER PLATFORM



Red Hat

OPENSHIFT ARCHITECTURE



Red Hat

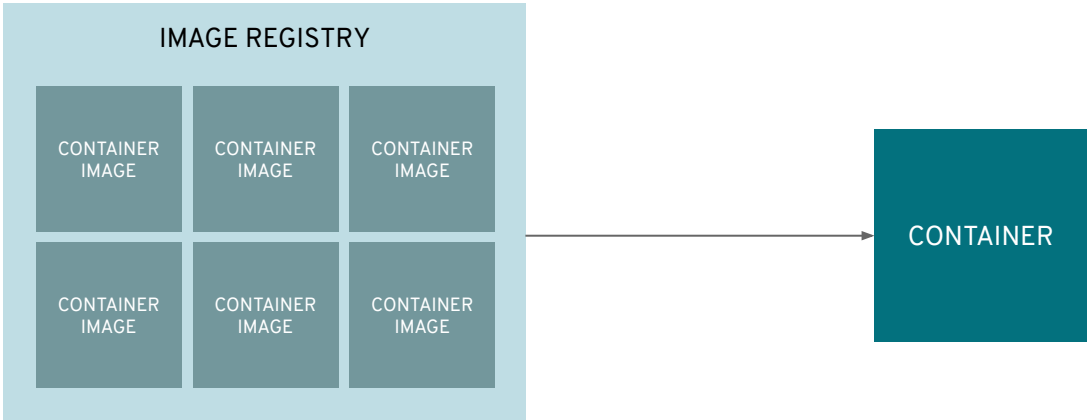
Cotainer Concepts Overview

Red Hat

A container is the smallest compute unit

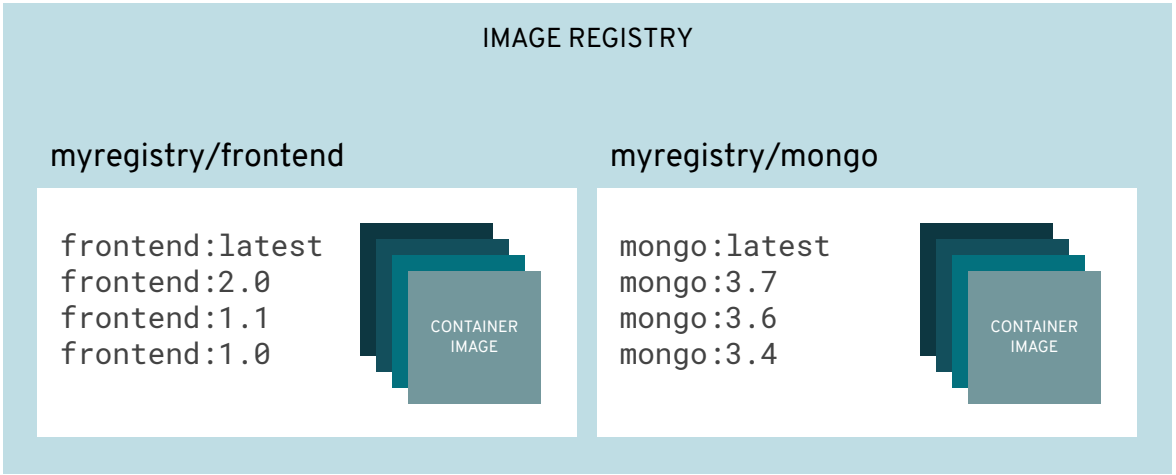
containers are created from container images

container images are stored in an image registry



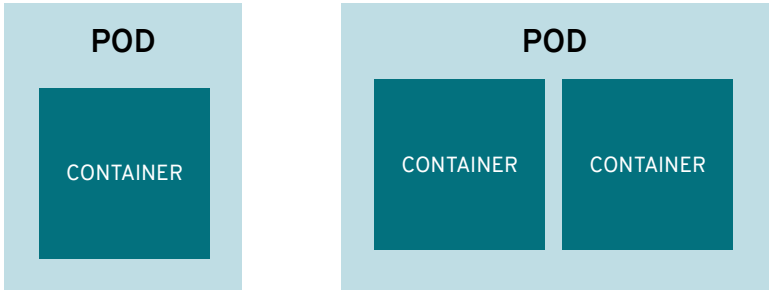
Red Hat

an image repository contains all versions of an image in the image registry



Red Hat

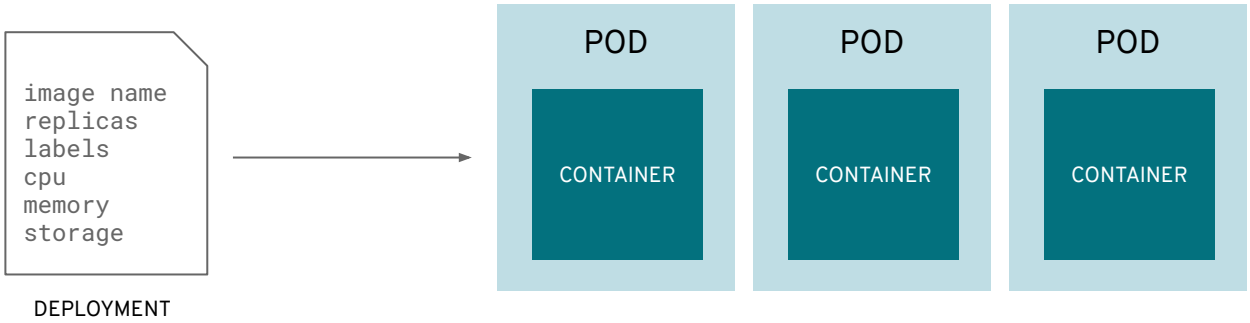
containers are wrapped in pods which are units of deployment and management



IP: 10.1.0.11

IP: 10.1.0.55

pods configuration is defined in a `deployment`



Red Hat

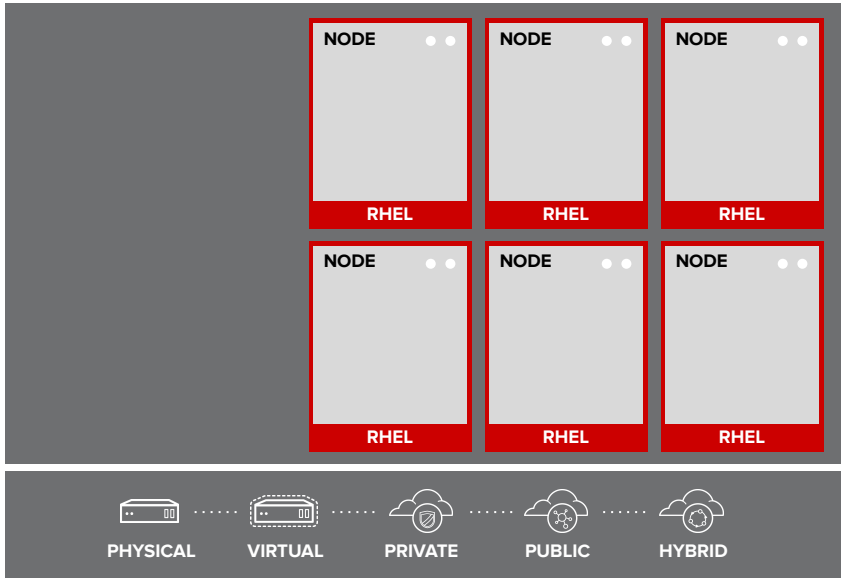
OpenShift Architecture

Red Hat

YOUR CHOICE OF INFRASTRUCTURE

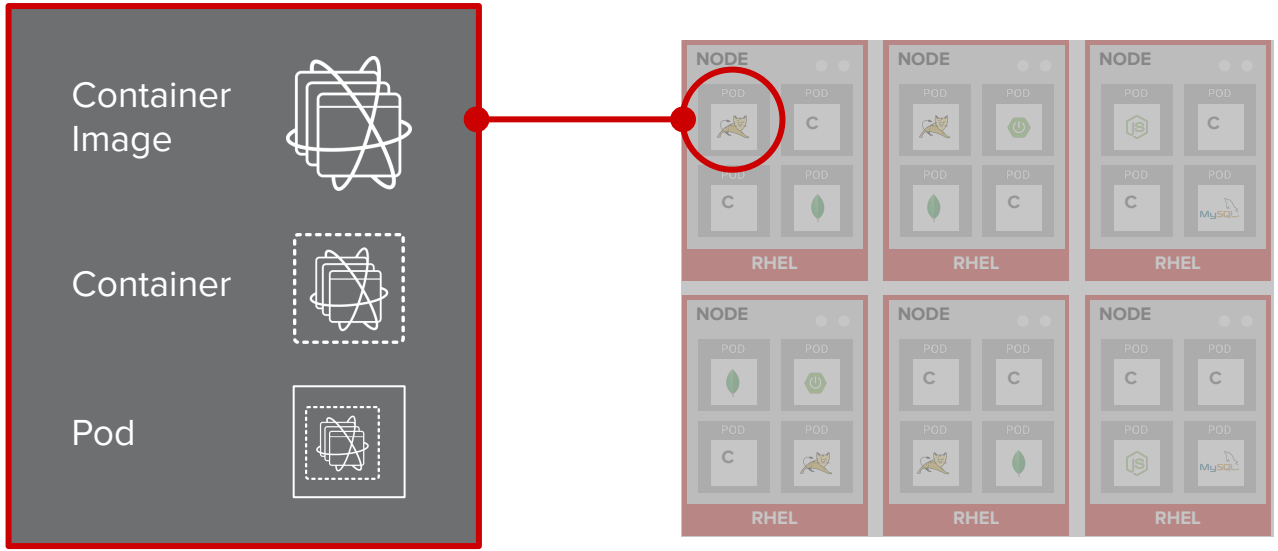
Red Hat

NODES RHEL INSTANCES WHERE APPS RUN



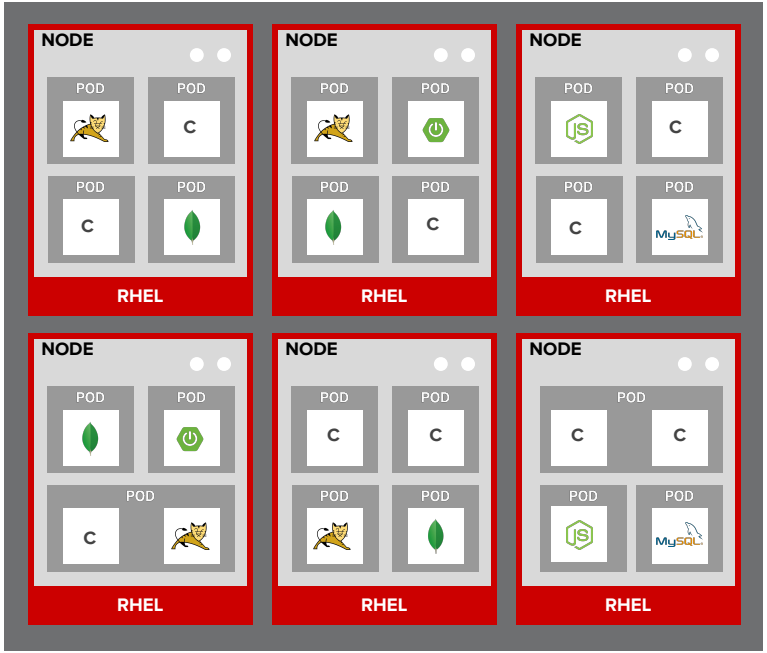
Red Hat

APPS RUN IN CONTAINERS



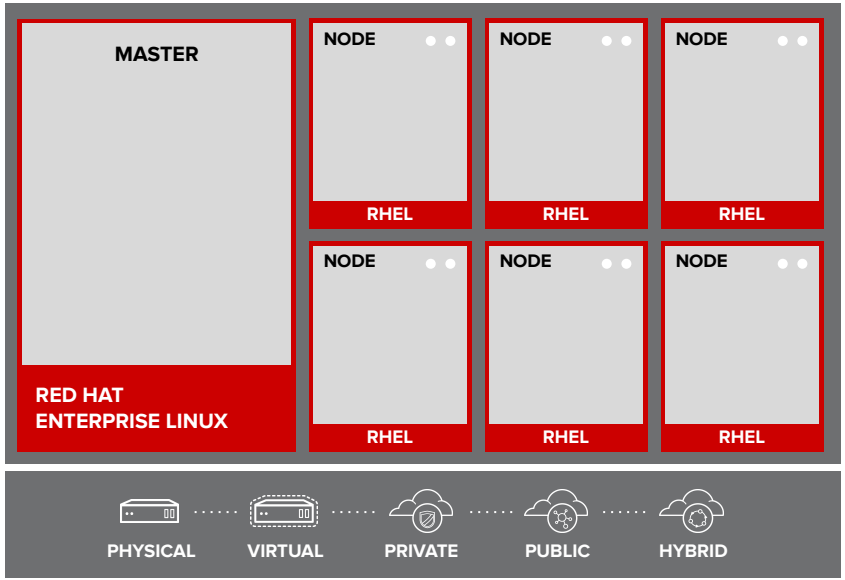
Red Hat

PODS ARE THE UNIT OF ORCHESTRATION



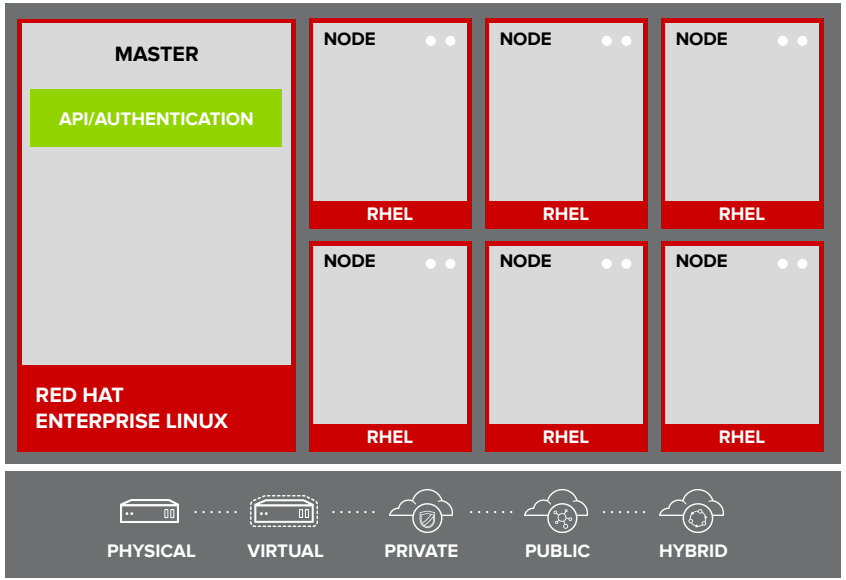
Red Hat

MASTERS ARE THE CONTROL PLANE



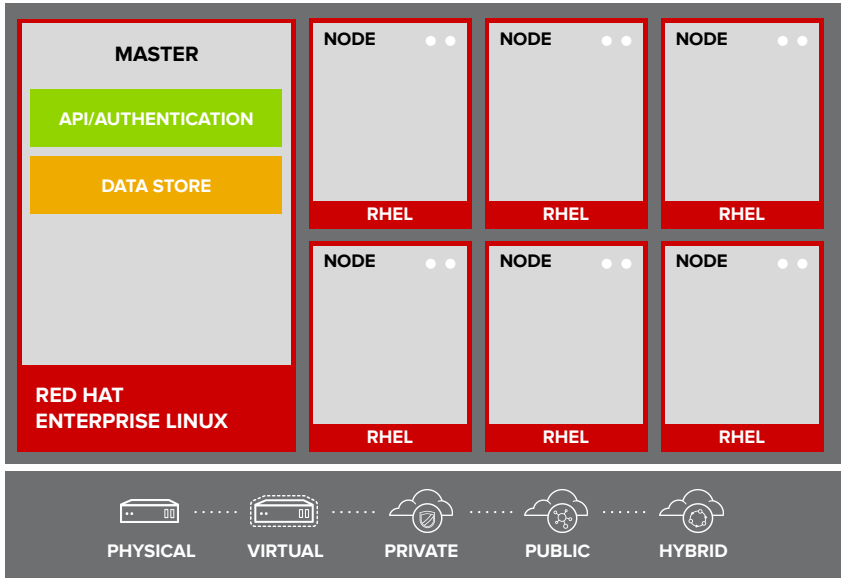
Red Hat

API AND AUTHENTICATION



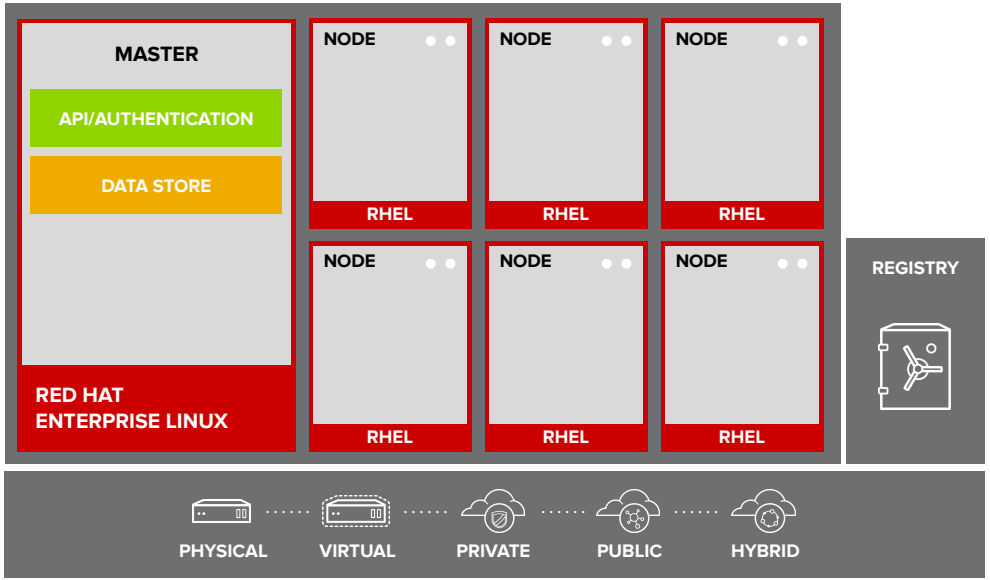
Red Hat

DESIRED AND CURRENT STATE



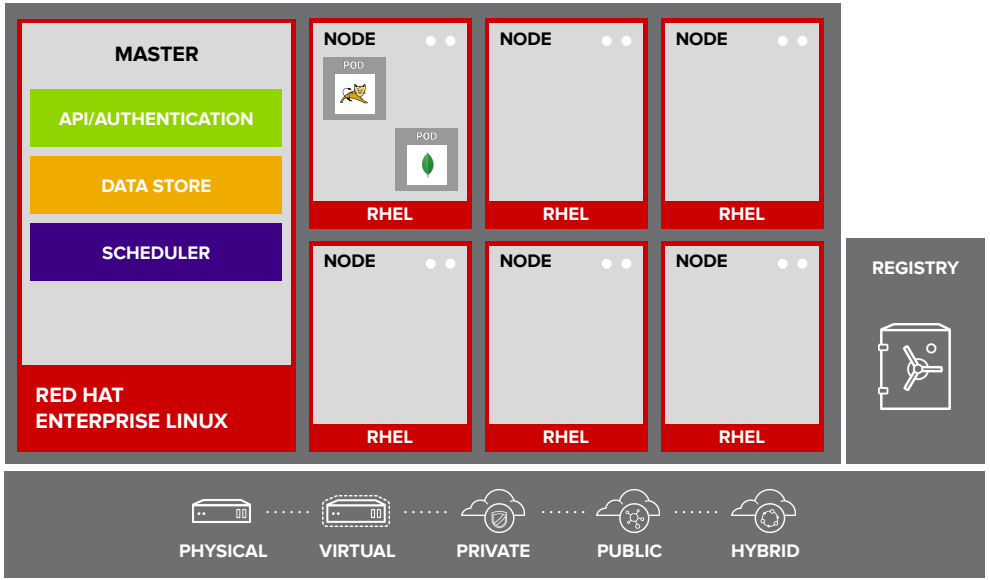
Red Hat

INTEGRATED CONTAINER REGISTRY



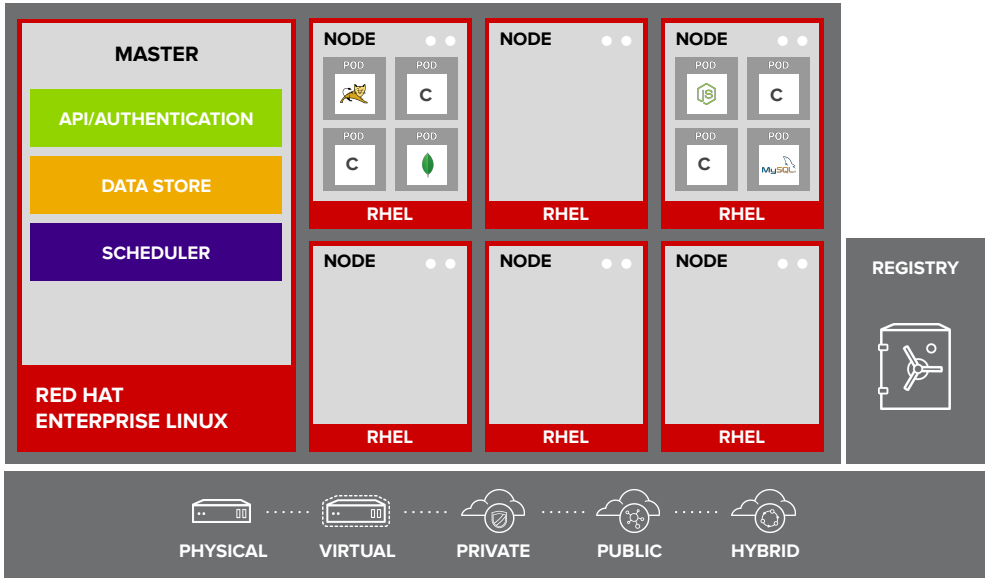
Red Hat

ORCHESTRATION AND SCHEDULING



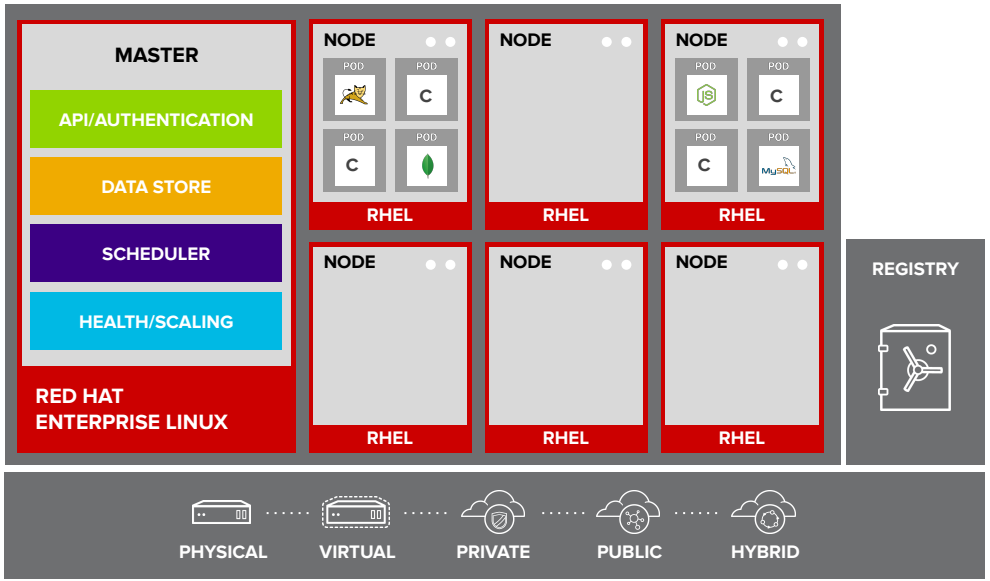
Red Hat

PLACEMENT BY POLICY

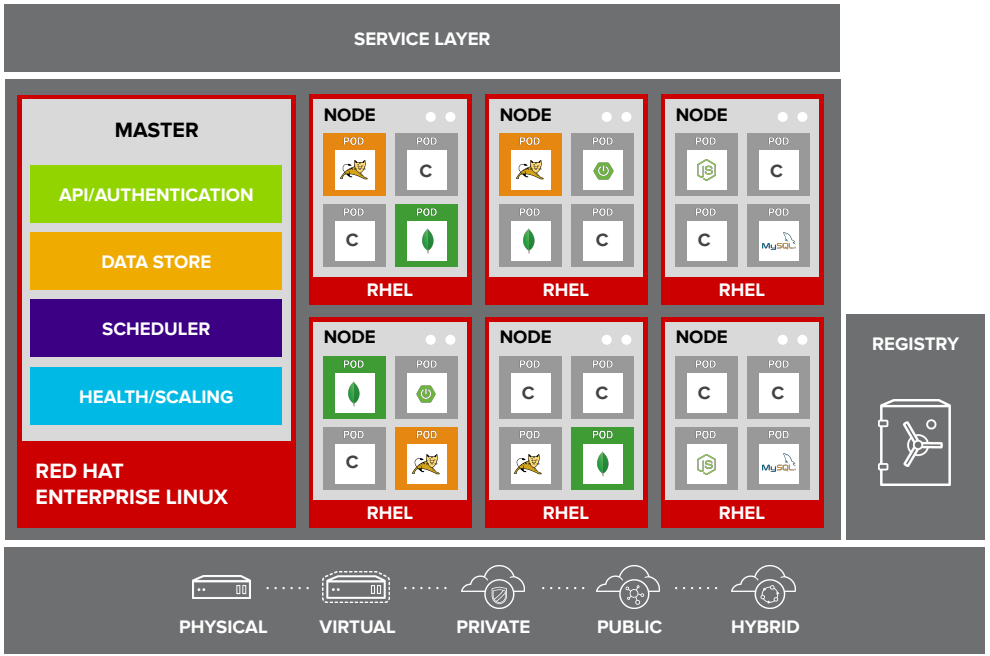


Red Hat

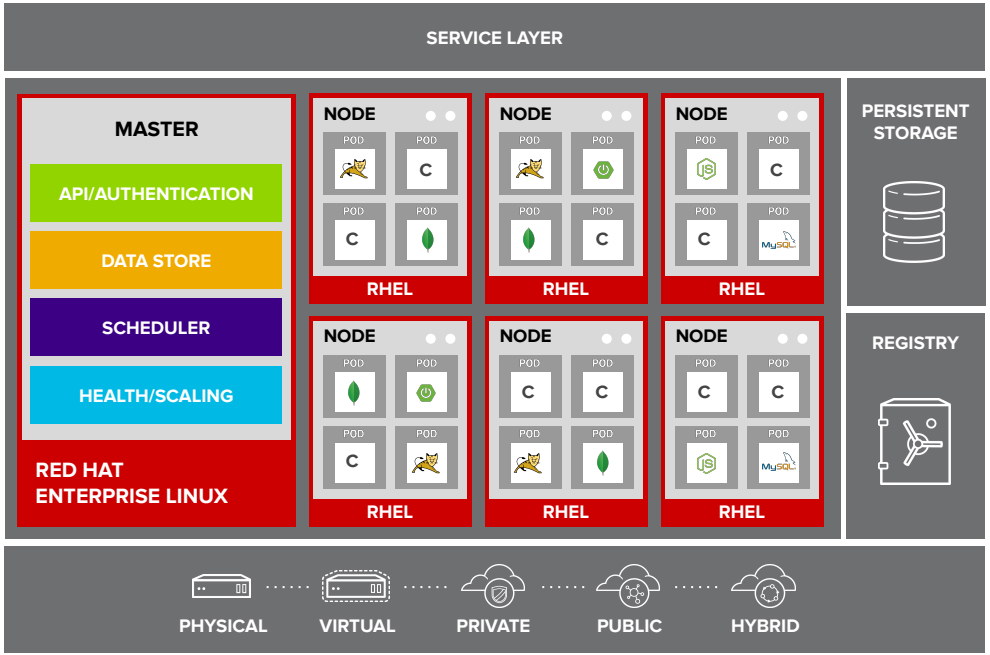
AUTOSCALING PODS



SERVICE DISCOVERY

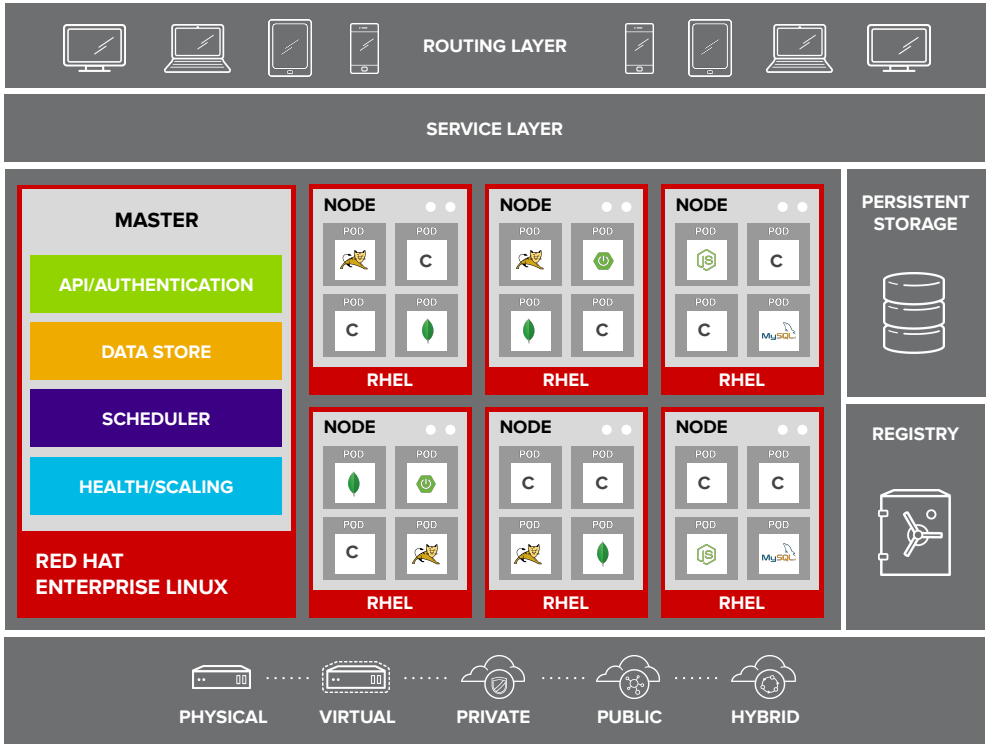


PERSISTENT DATA IN CONTAINERS



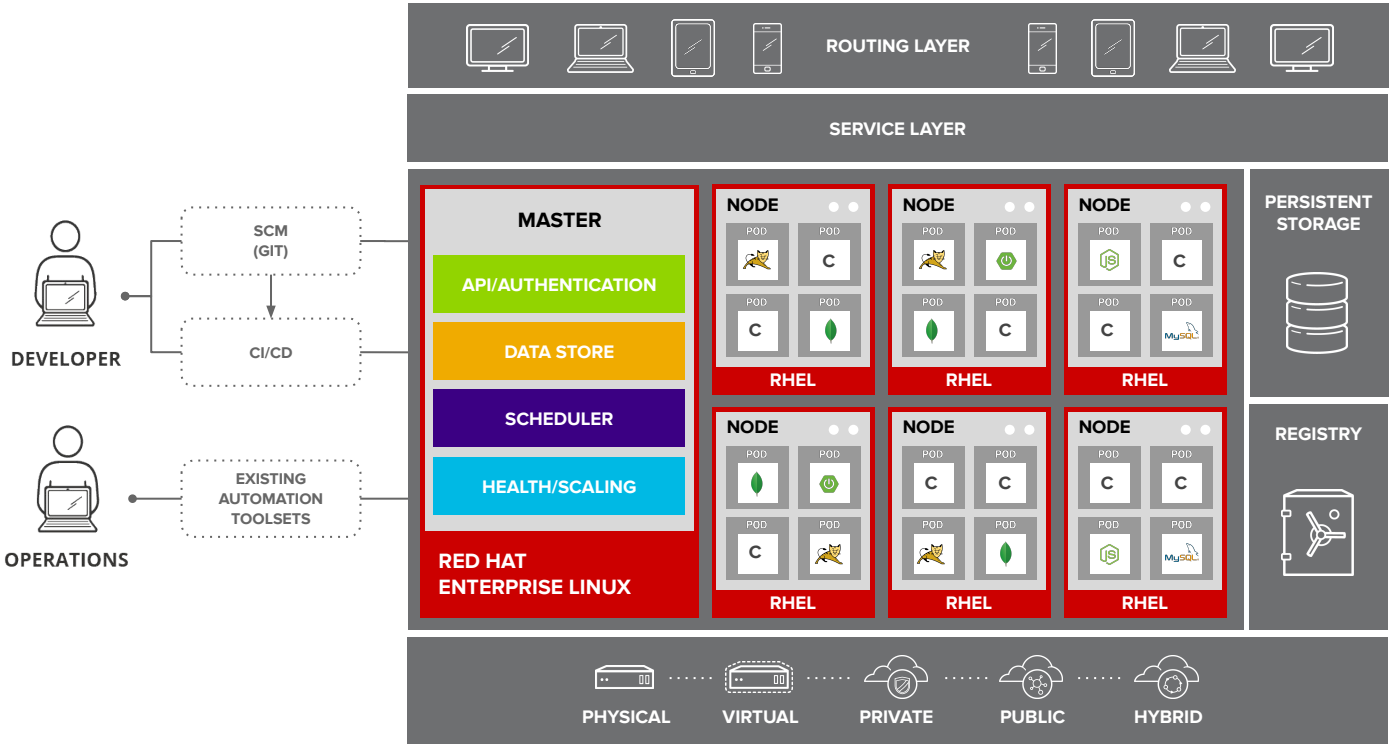
Red Hat

ROUTING AND LOAD-BALANCING



Red Hat

ACCESS VIA WEB, CLI, IDE AND API

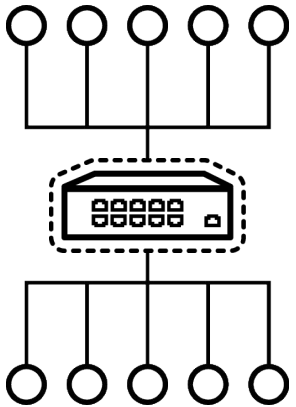


Red Hat

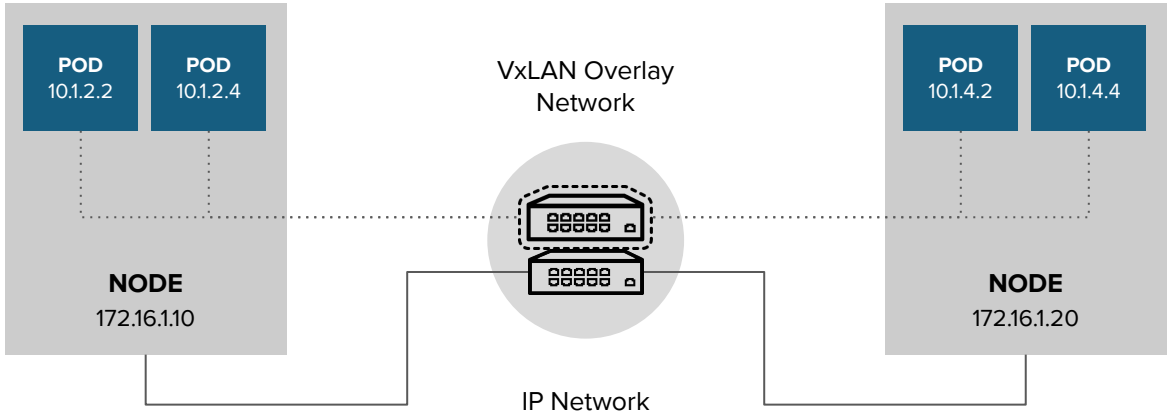
Networking

OPENSHIFT NETWORKING

- Built-in internal DNS to reach services by name
- Split DNS is supported via DNSmasq
  - Master answers DNS queries for internal services
  - Other name servers serve the rest of the queries
- Software Defined Networking (SDN) for a unified cluster network to enable pod-to-pod communication
- OpenShift follows the Kubernetes Container Networking Interface (CNI) plug-in model

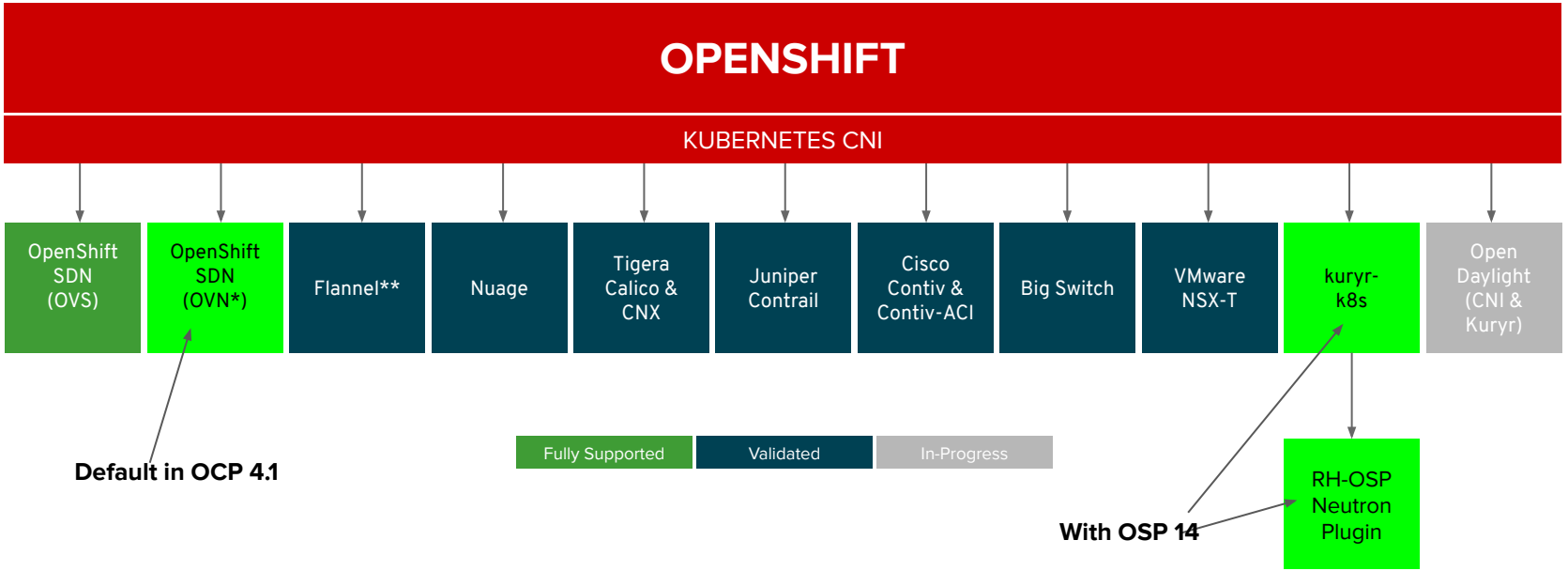


OPENSHIFT NETWORKING



Red Hat

OPENSHIFT NETWORK PLUGINS



Red Hat

OPENSHIFT SDN

FLAT NETWORK

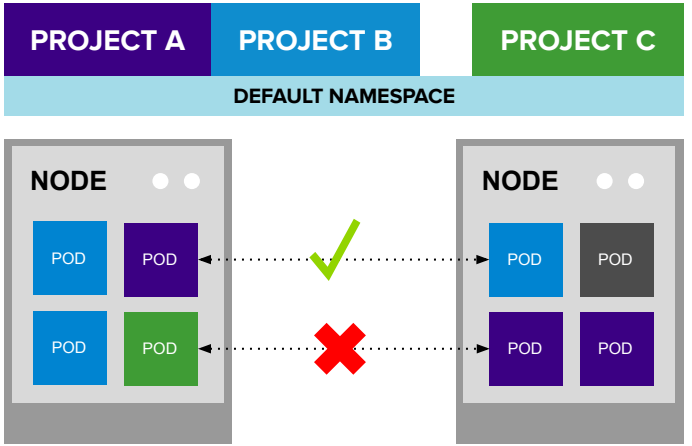
- All pods can communicate with each other across projects

MULTI-TENANT NETWORK

- Project-level network isolation
- Multicast support
- Egress network policies

NETWORK POLICY (Default)

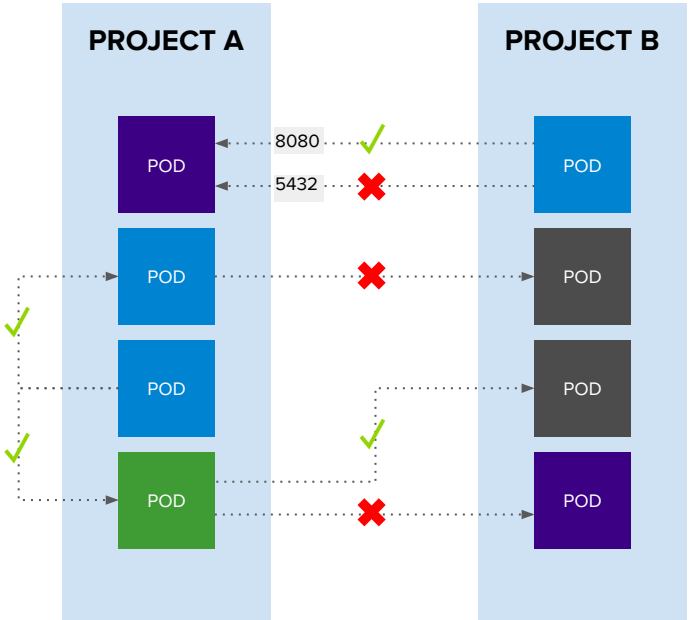
- Granular policy-based isolation



Multi-Tenant Network

Red Hat

OPENSHIFT SDN - NETWORK POLICY



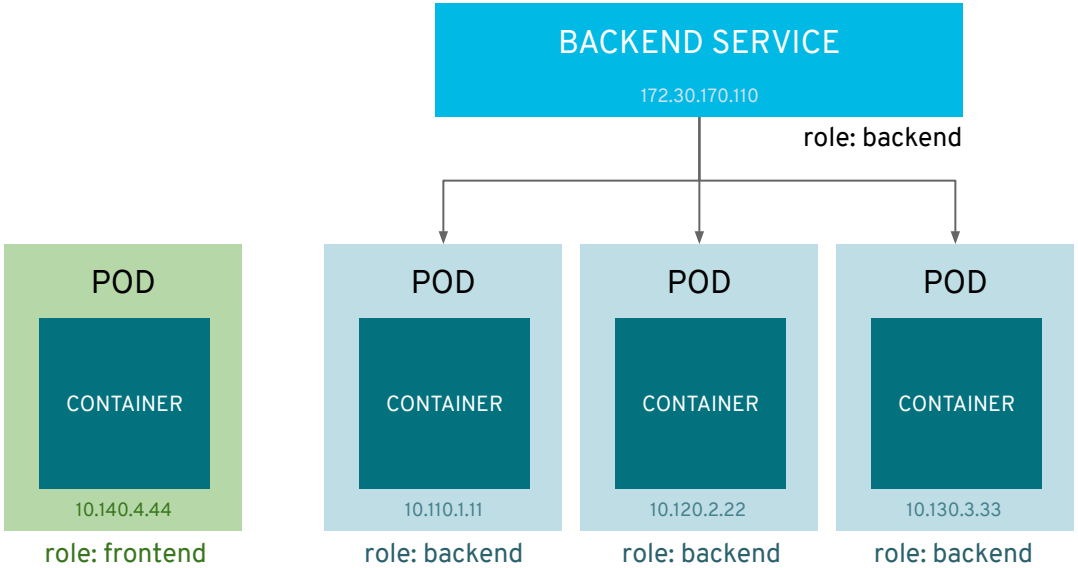
Example Policies

- Allow all traffic inside the project
- Allow traffic from green to gray
- Allow traffic to purple on 8080

```
apiVersion: extensions/v1beta1
kind: NetworkPolicy
metadata:
  name: allow-to-purple-on-8080
spec:
  podSelector:
    matchLabels:
      color: purple
  ingress:
  - ports:
    - protocol: tcp
      port: 8080
```

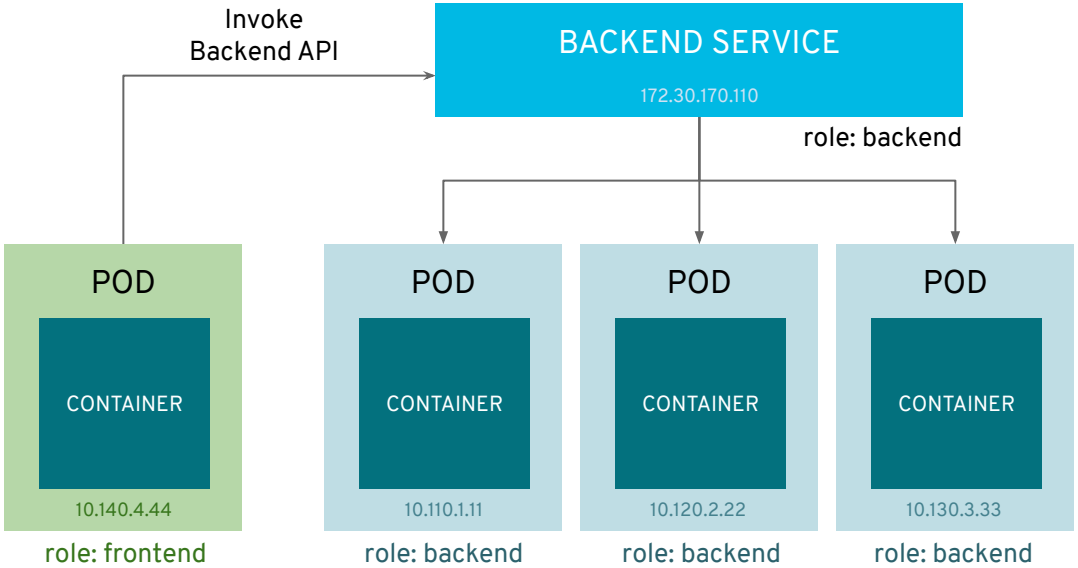

Red Hat

services provide internal load-balancing and service discovery across pods



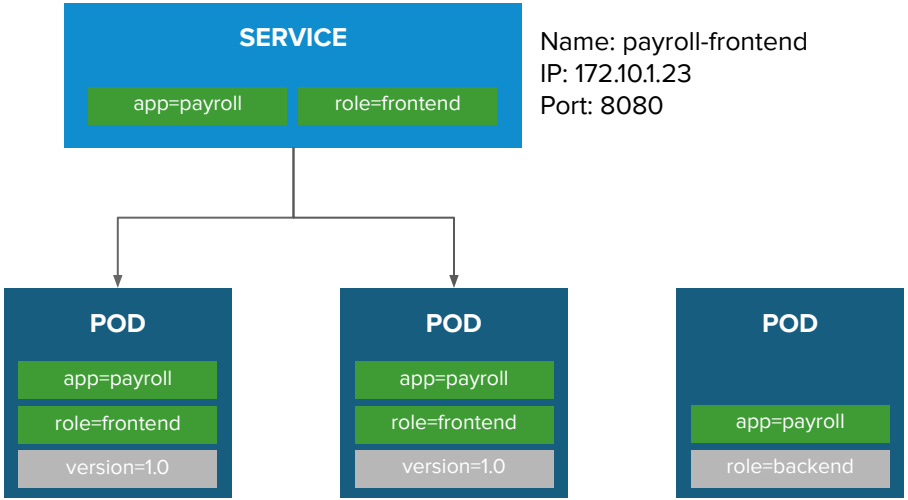
Red Hat

apps can talk to each other via services



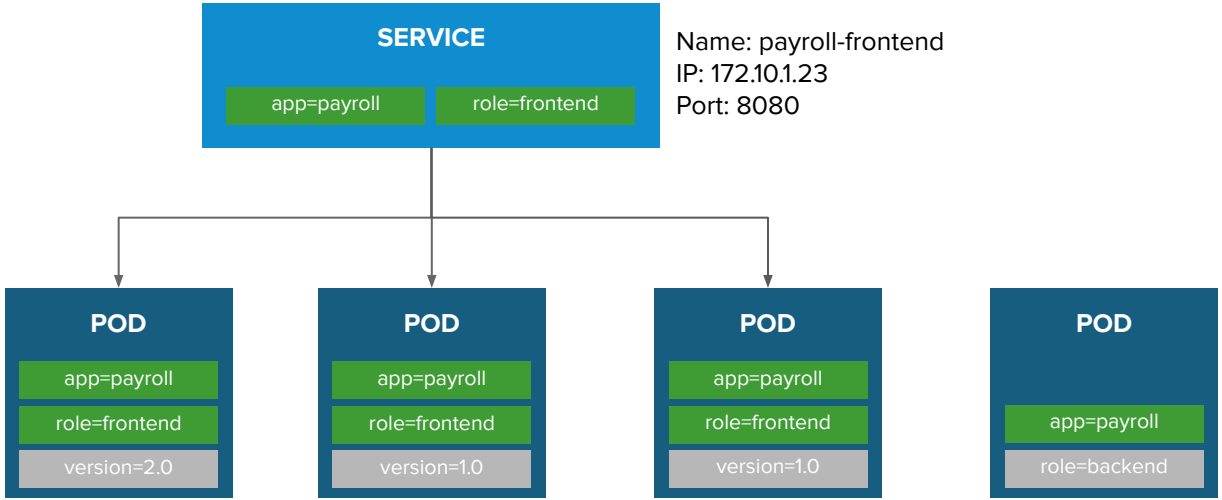
Red Hat

BUILT-IN SERVICE DISCOVERY
INTERNAL LOAD-BALANCING



Red Hat

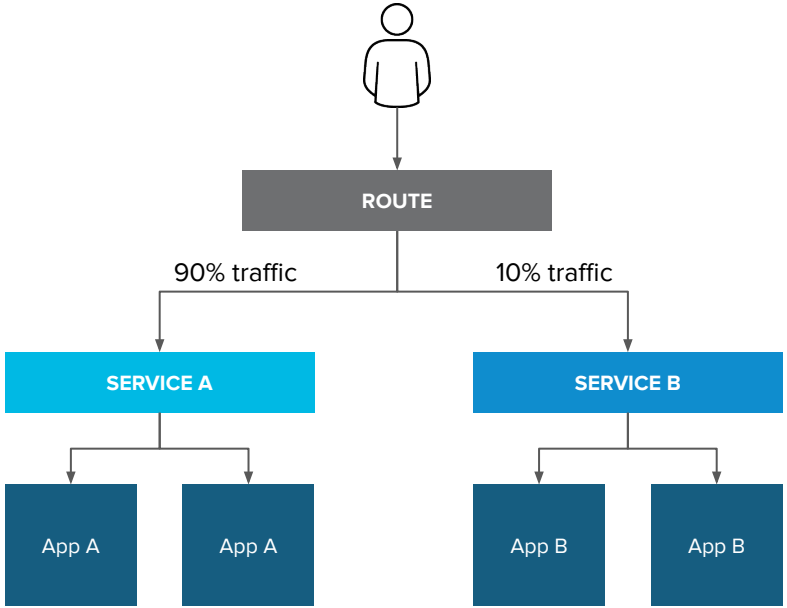
BUILT-IN SERVICE DISCOVERY
INTERNAL LOAD-BALANCING



Red Hat

ROUTE SPLIT TRAFFIC

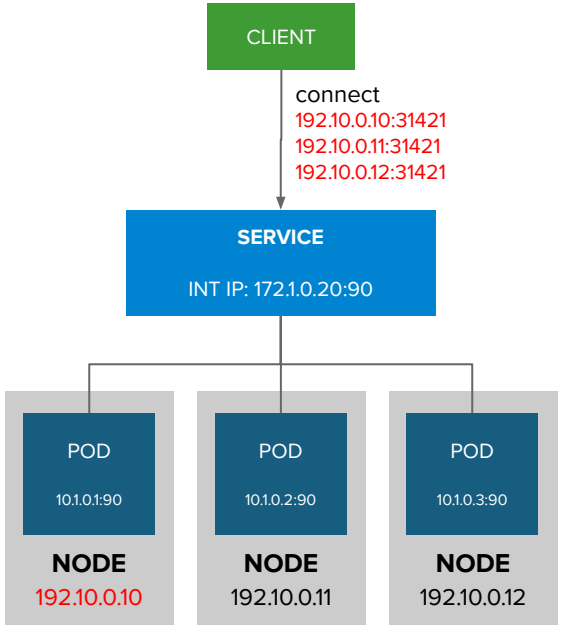
Split Traffic Between Multiple Services For A/B Testing, Blue/Green and Canary Deployments



Red Hat

EXTERNAL TRAFFIC TO A SERVICE ON A RANDOM PORT WITH NODEPORT

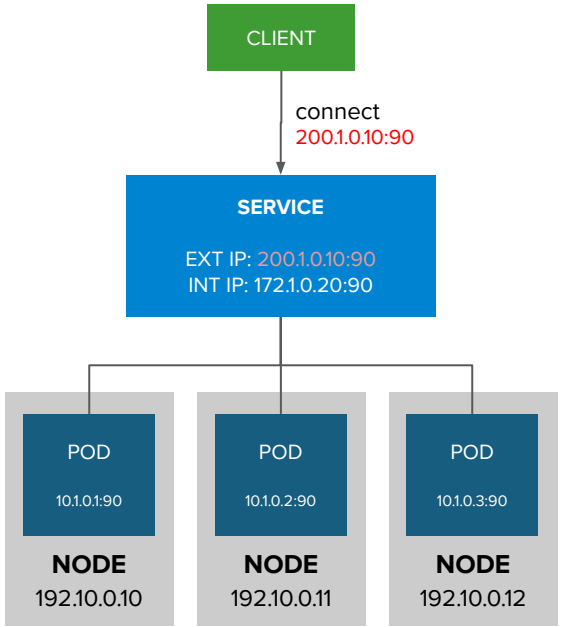
- NodePort binds a service to a unique port on all the nodes
- Traffic received on any node redirects to a node with the running service
- Ports in 30K-60K range which usually differs from the service
- Firewall rules must allow traffic to all nodes on the specific port



Red Hat

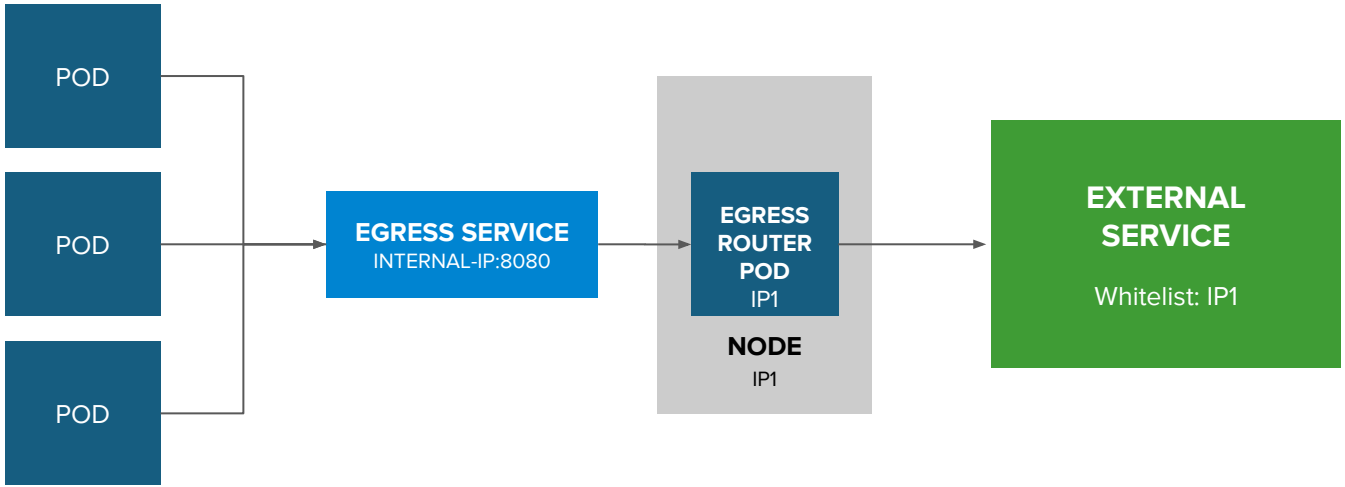
EXTERNAL TRAFFIC TO A SERVICE ON ANY PORT WITH INGRESS

- Access a service with an external IP on any TCP/UDP port, such as
  - Databases
  - Message Brokers
- Automatic IP allocation from a predefined pool using Ingress IP Self-Service
- IP failover pods provide high availability for the IP pool



Red Hat

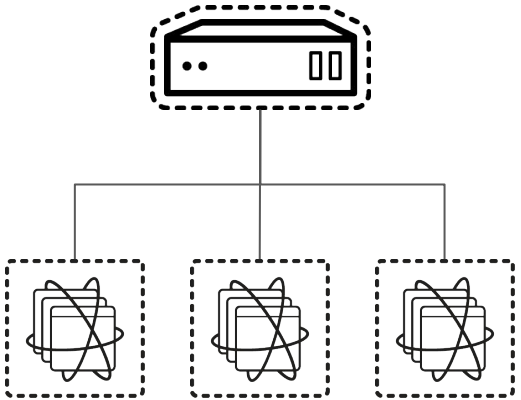
CONTROL OUTGOING TRAFFIC SOURCE IP WITH EGRESS ROUTER



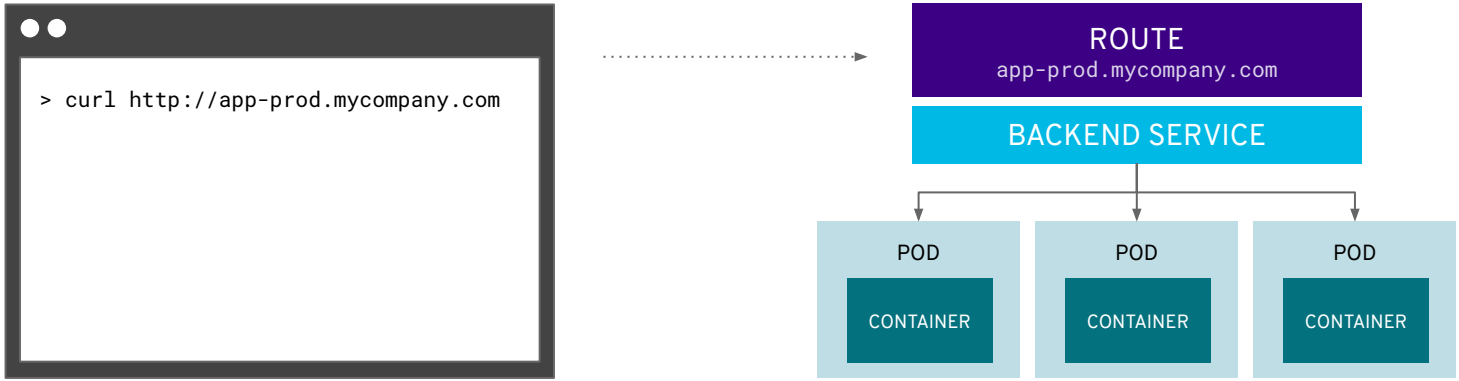
Red Hat

ROUTING AND EXTERNAL LOAD-BALANCING

- Pluggable routing architecture
  - HAProxy Router
  - F5 Router
- Multiple-routers with traffic sharding
- Router supported protocols
  - HTTP/HTTPS
  - WebSockets
  - TLS with SNI
- Non-standard ports via cloud load-balancers, external IP, and NodePort

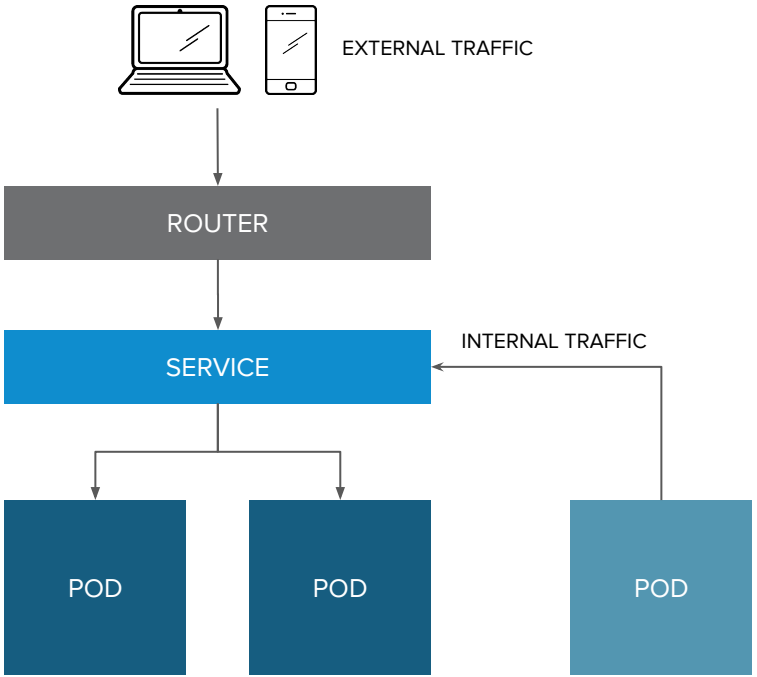


routes add services to the external load-balancer and provide readable urls for the app



Red Hat

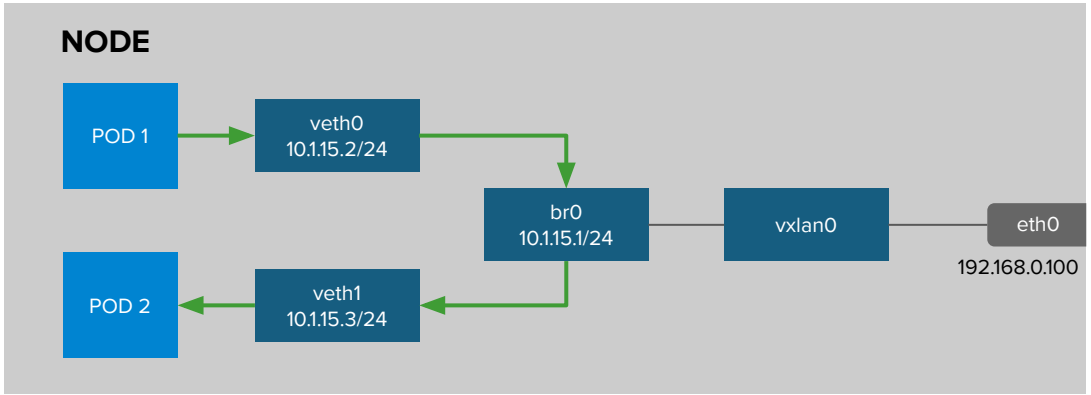
ROUTE EXPOSES SERVICES EXTERNALLY



Red Hat

OPENSHIFT SDN - OVS PACKET FLOW

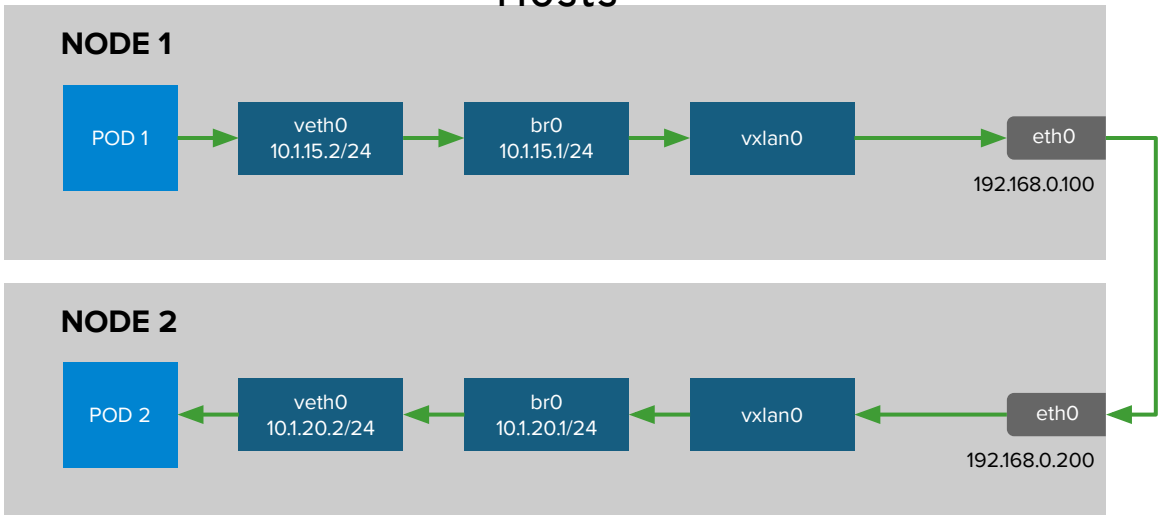
Container to Container on the Same Host



Red Hat

OPENSHIFT SDN - OVS PACKET FLOW

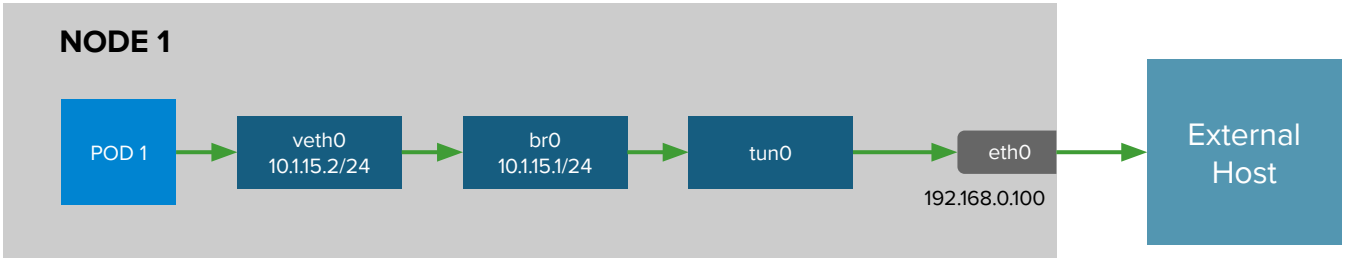
Container to Container on the Different Hosts



Red Hat

OPENSHIFT SDN - OVS PACKET FLOW

Container Connects to External Host

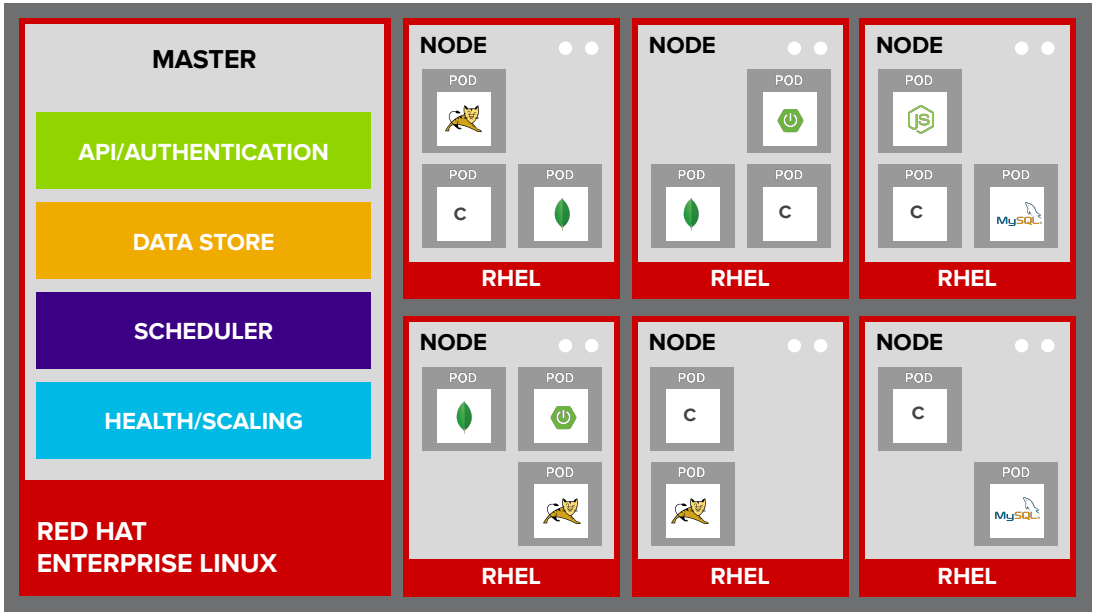


Red Hat

OpenShift Monitoring / Clustering

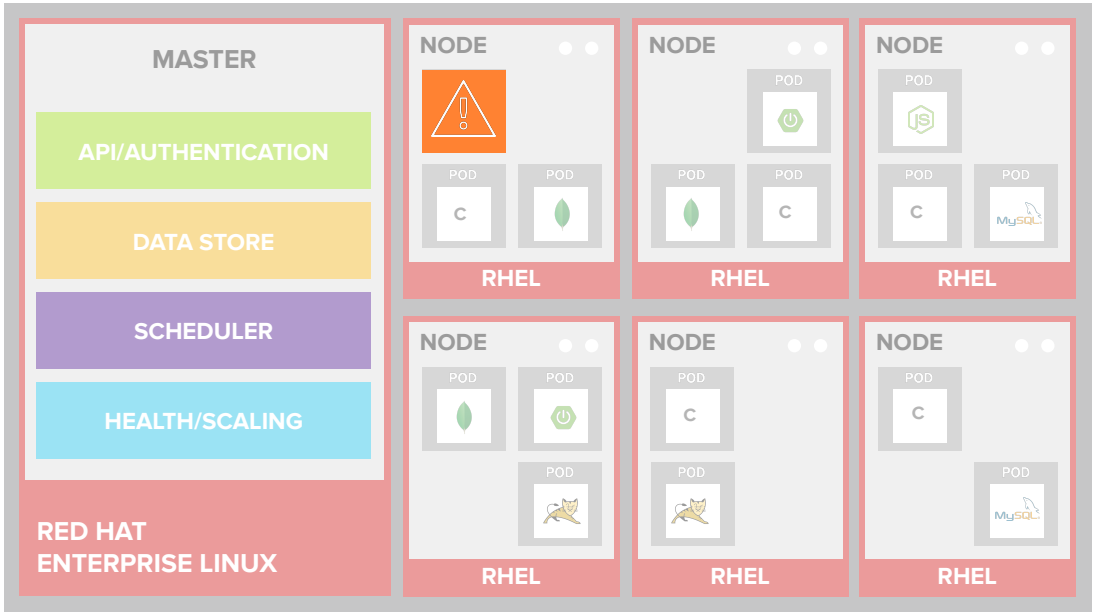
Red Hat

AUTO-HEALING FAILED PODS



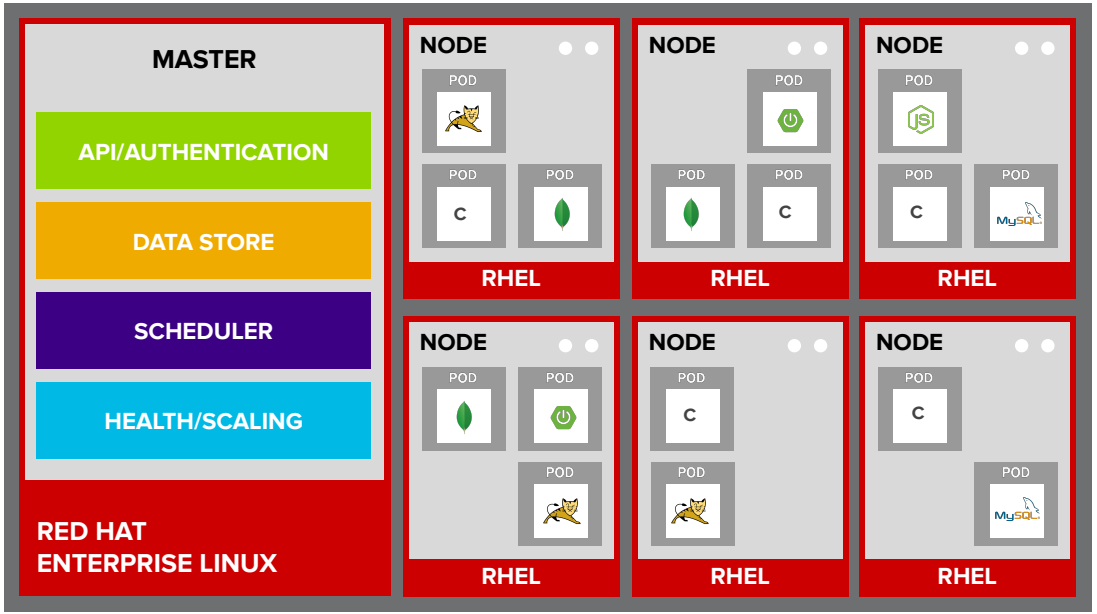
Red Hat

AUTO-HEALING FAILED CONTAINERS



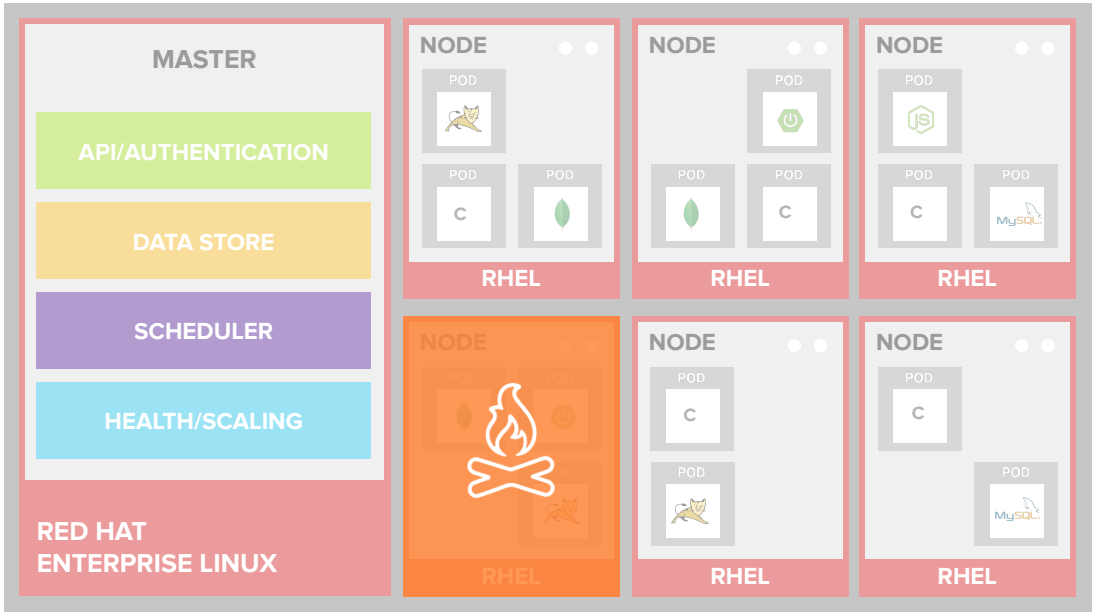
Red Hat

AUTO-HEALING FAILED CONTAINERS



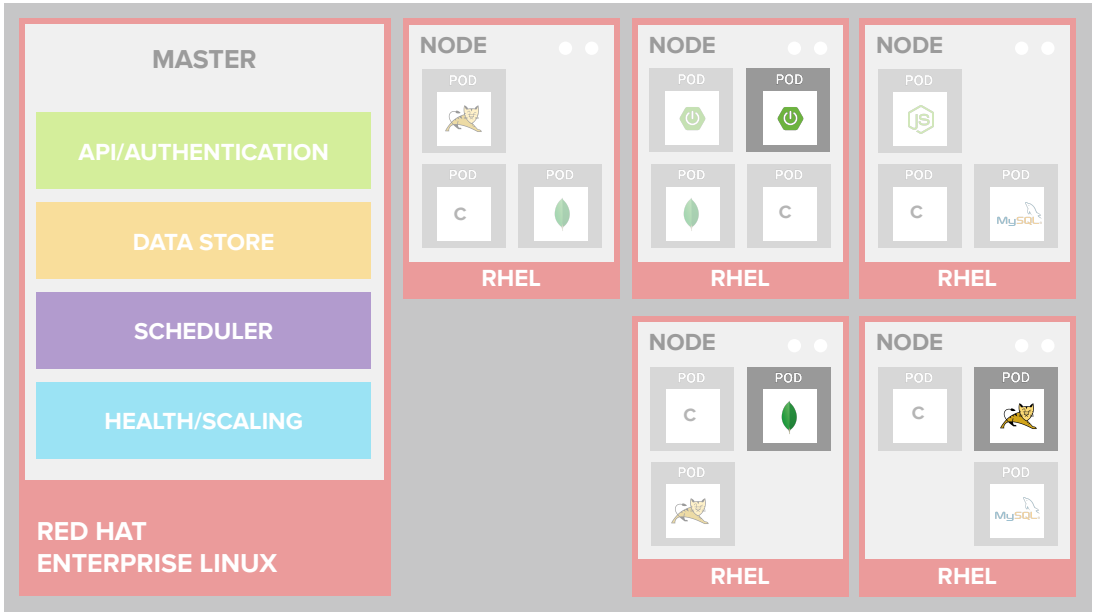
Red Hat

AUTO-HEALING FAILED CONTAINERS



Red Hat

AUTO-HEALING FAILED CONTAINERS



Red Hat

OpenShift persistent Storage

Red Hat

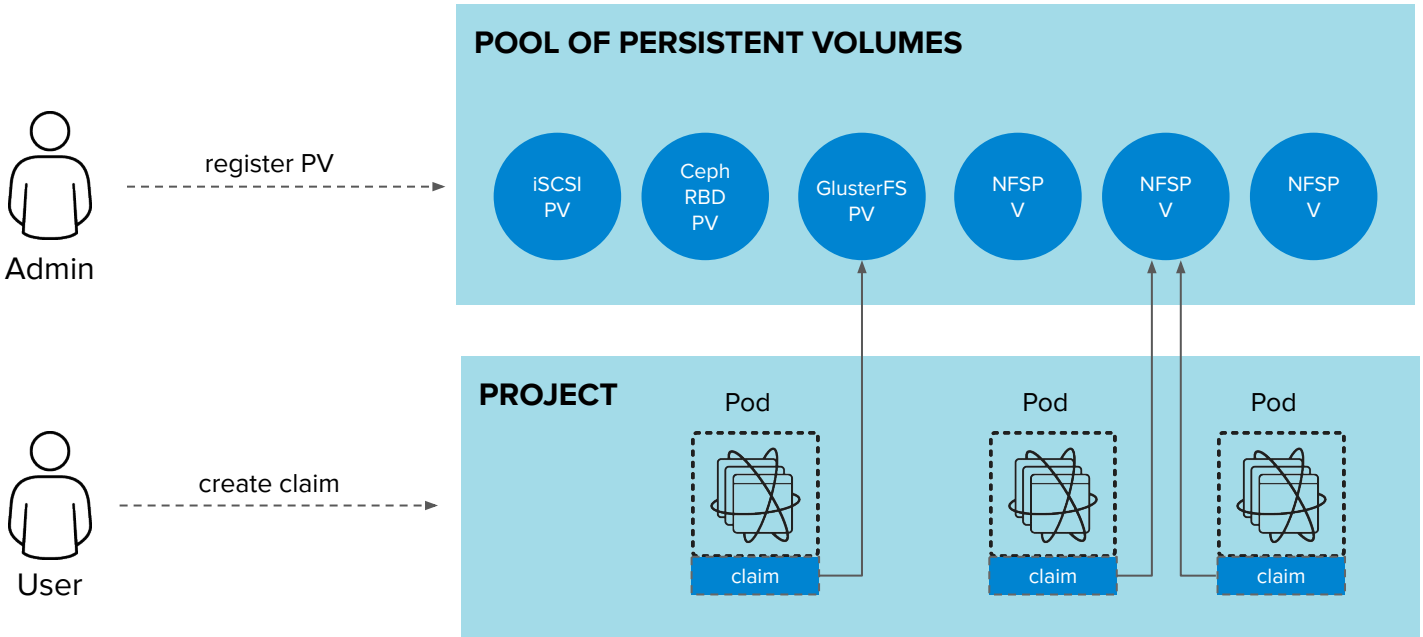
PERSISTENT STORAGE

- Persistent Volume (PV) is tied to a piece of network storage
- Provisioned by an administrator (static or dynamically)
- Allows admins to describe storage and users to request storage
- Assigned to pods based on the requested size, access mode, labels and type

NFS	OpenStack Cinder	iSCSI	Azure Disk	AWS EBS	FlexVolume
GlusterFS	Ceph RBD	Fiber Channel	Azure File	GCE Persistent Disk	VMWare vSphere VMDK
		NetApp Trident*	Container Storage Interface (CSI)**		

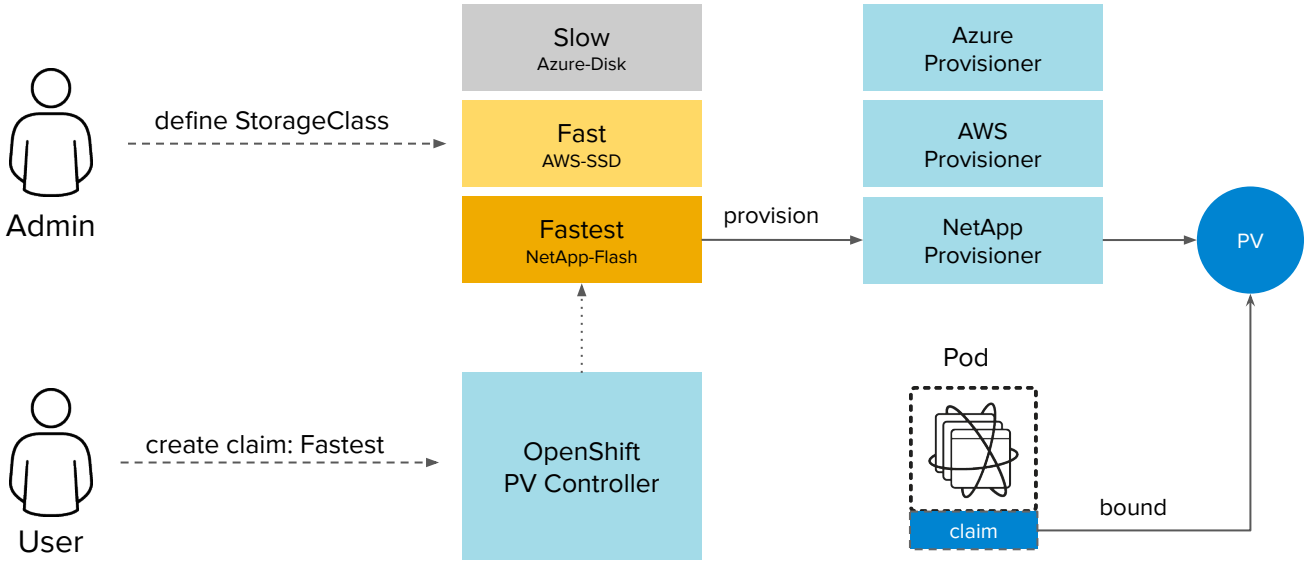
* Shipped and supported by NetApp via TSANet
** Tech Preview

PERSISTENT STORAGE



Red Hat

DYNAMIC VOLUME PROVISIONING



Red Hat

Thank you !