

# Introduction to OKD

**Alan Adi Prastyo** from I3



# About Me

## Alan Adi Prastyo

Senior Consultant at Inovasi Informatika Indonesia (I3)

Linux Geek, Kubernetes & Openshift Enthusiast

RHCSA, RHCE, RHCSA in Openstack, Red Hat Certified Specialist in Openshift Administration, MTCNA, Certified Openstack Administration (COA), DevOps Foundation Certified, 3Scale API Management.

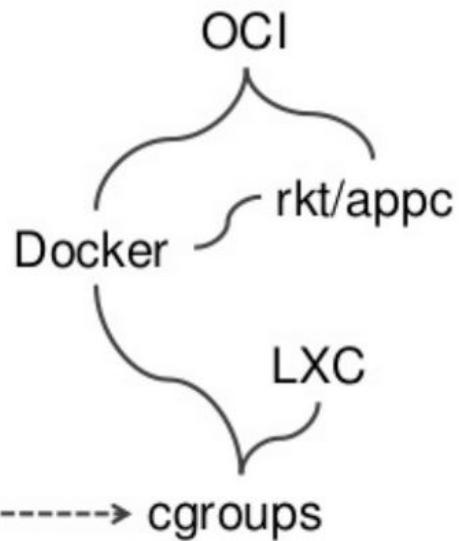
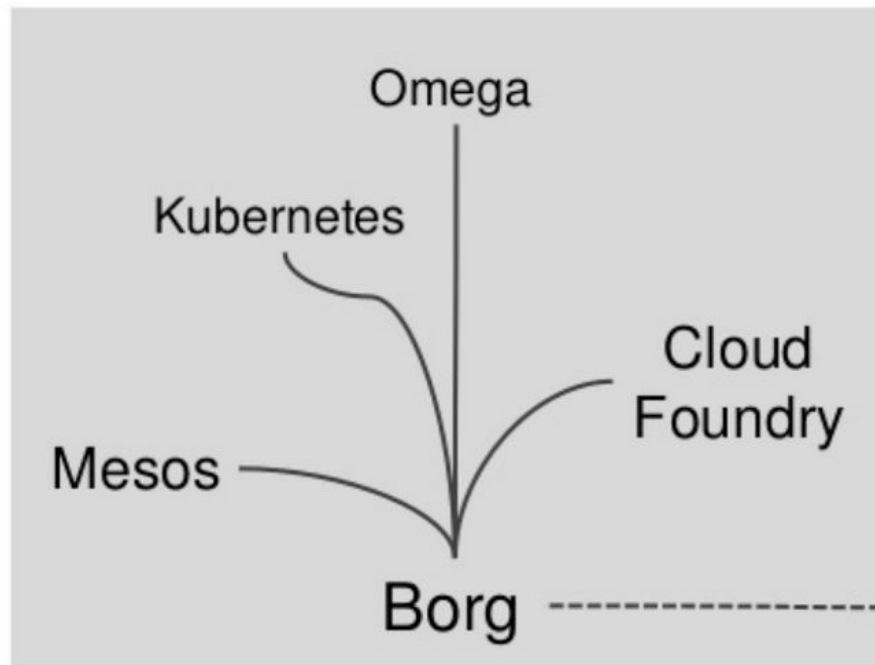


# Learning Objectives

- Discuss OKD
- Learn Basic OKD Terminology
- Learn Architecture OKD
- Learn Installation and Configuration tools



# Borg Heritage



# What is OKD?



OKD is a **distribution of Kubernetes** optimized for continuous application development and multi-tenant deployment. OKD adds **developer and operations-centric** tools on top of Kubernetes to enable rapid application development, easy deployment and scaling, and long-term lifecycle maintenance for small and large teams. OKD is the upstream Kubernetes distribution embedded in Red Hat OpenShift.

Source: <https://www.okd.io/#v3>

# OKD Features

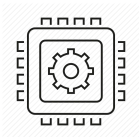
Self-Service



Multi-language



Automation



Collaboration



Multi-tenant



Web-scale



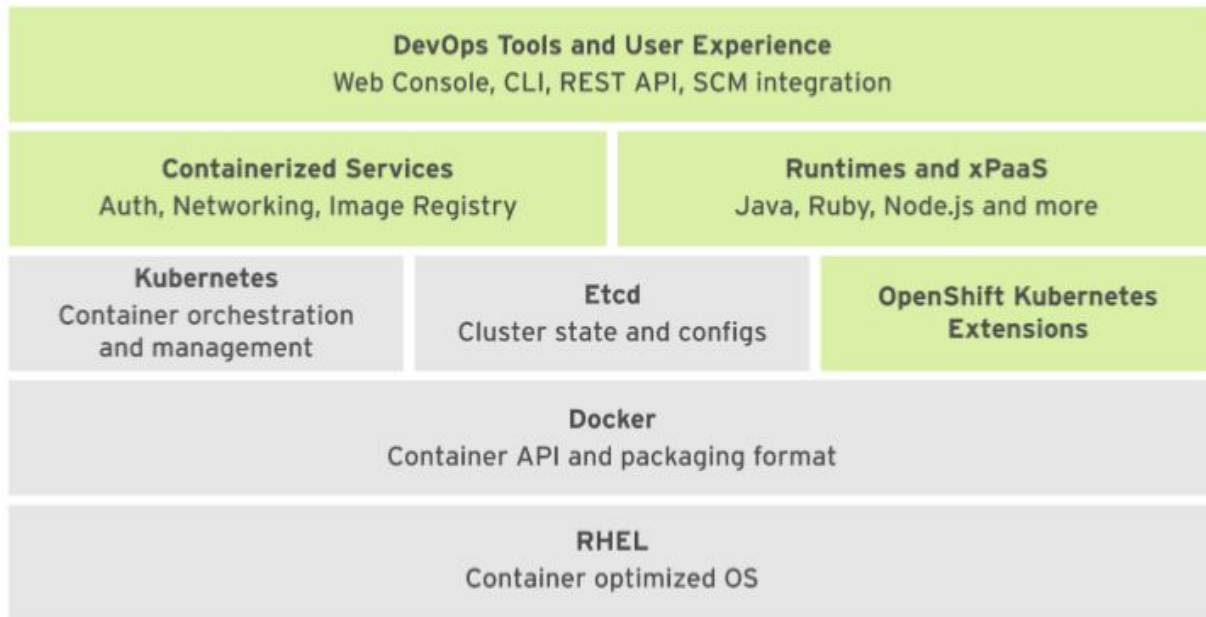
Open source



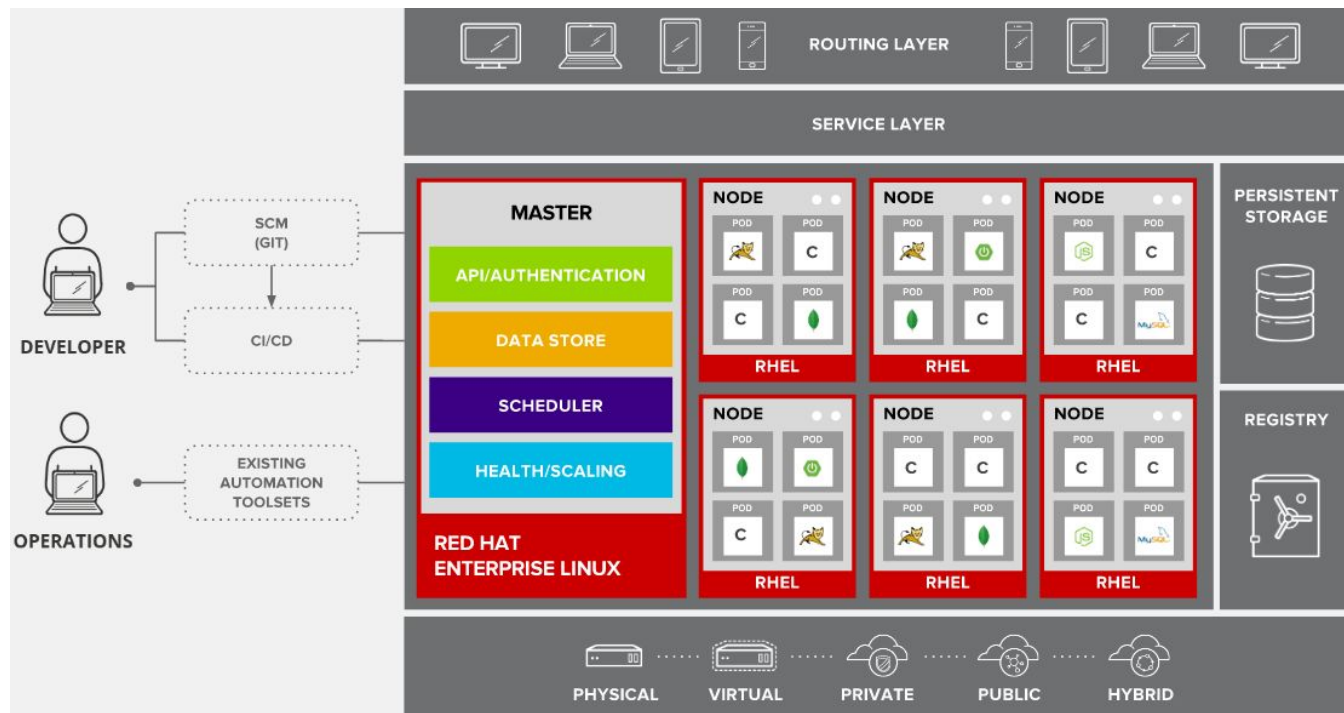
Standard-based



# OKD Architecture

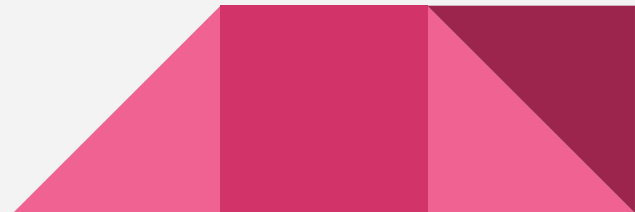


# OKD Architecture





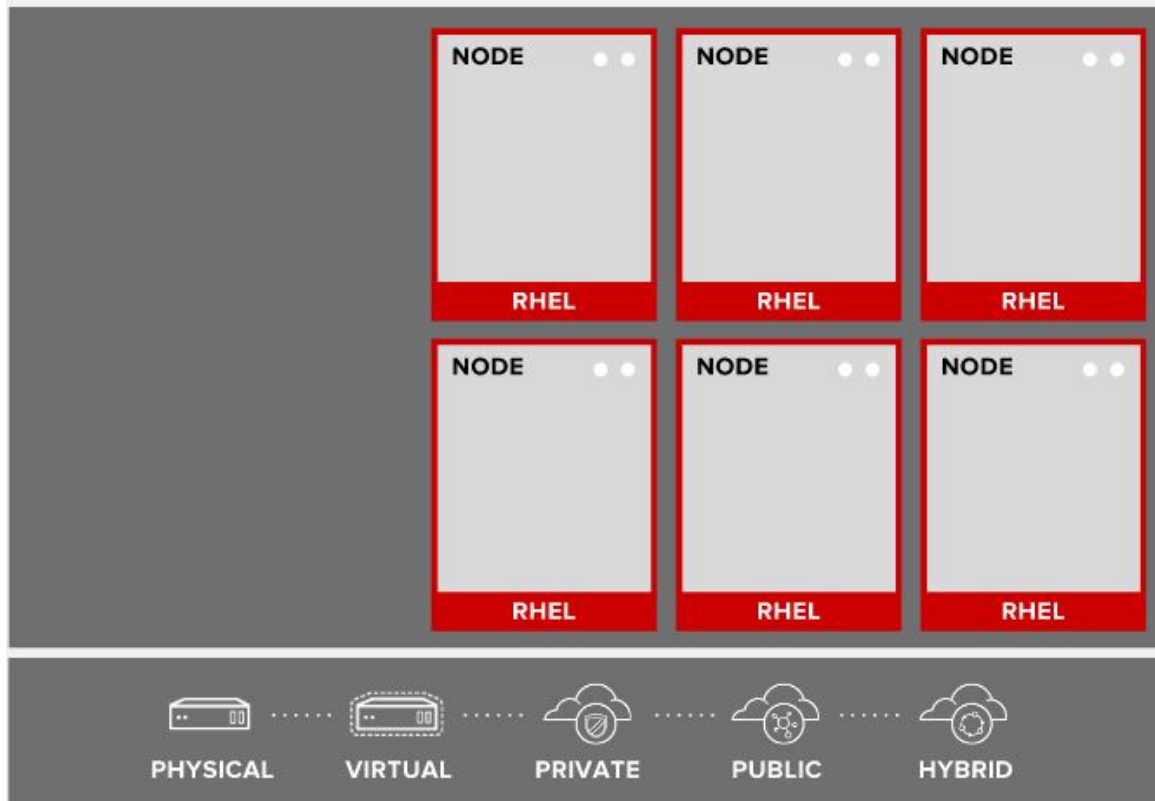
# OKD Architecture



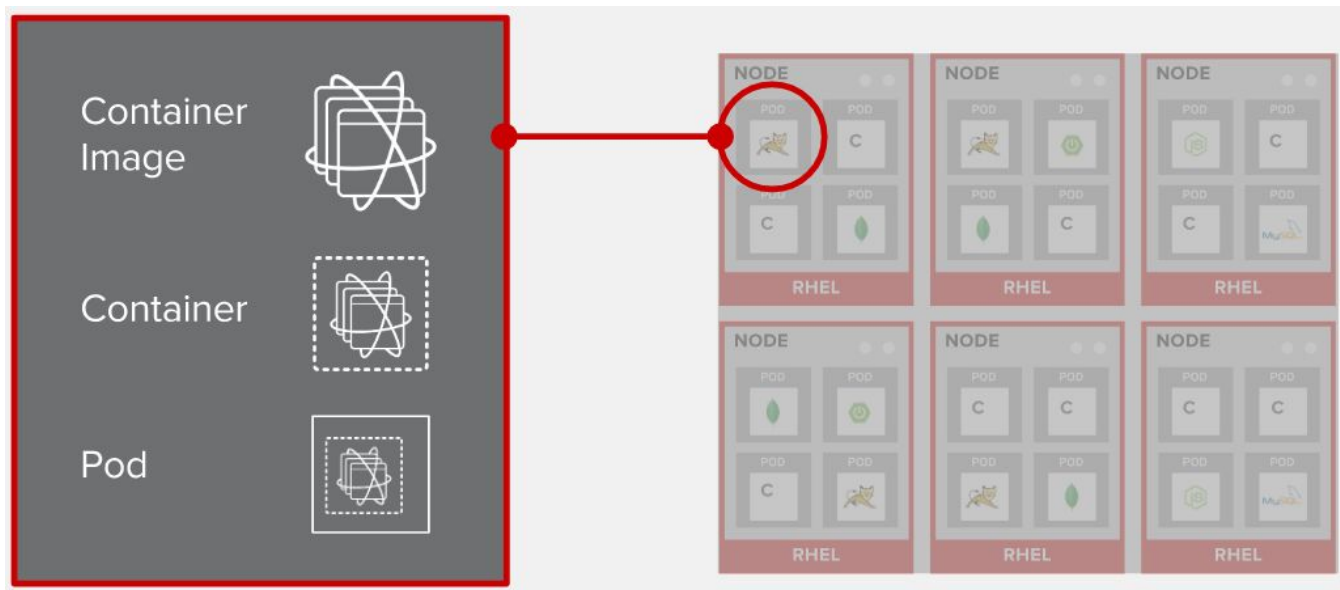
# YOUR CHOICE OF INFRASTRUCTURE



# NODES INSTANCES WHERE APPS RUN



# APPS RUN IN CONTAINERS



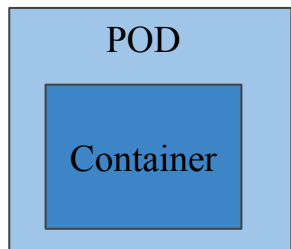
# A Container is the smallest compute unit



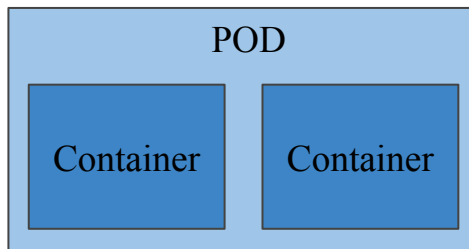
# Container are created from container images



# POD



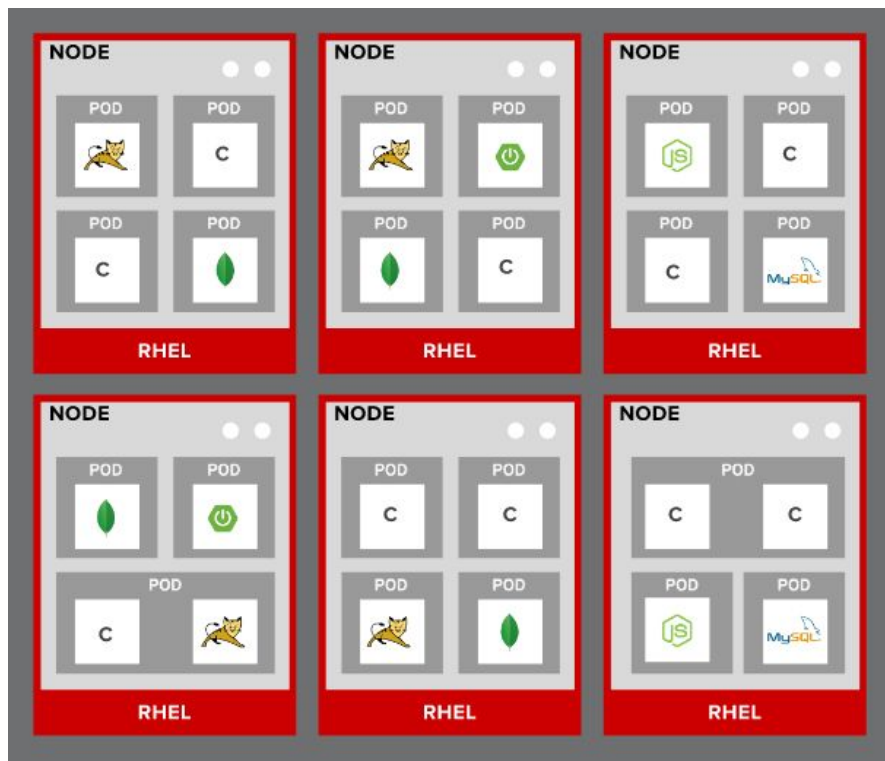
IP: 10.0.1.20



IP: 10.0.1.30

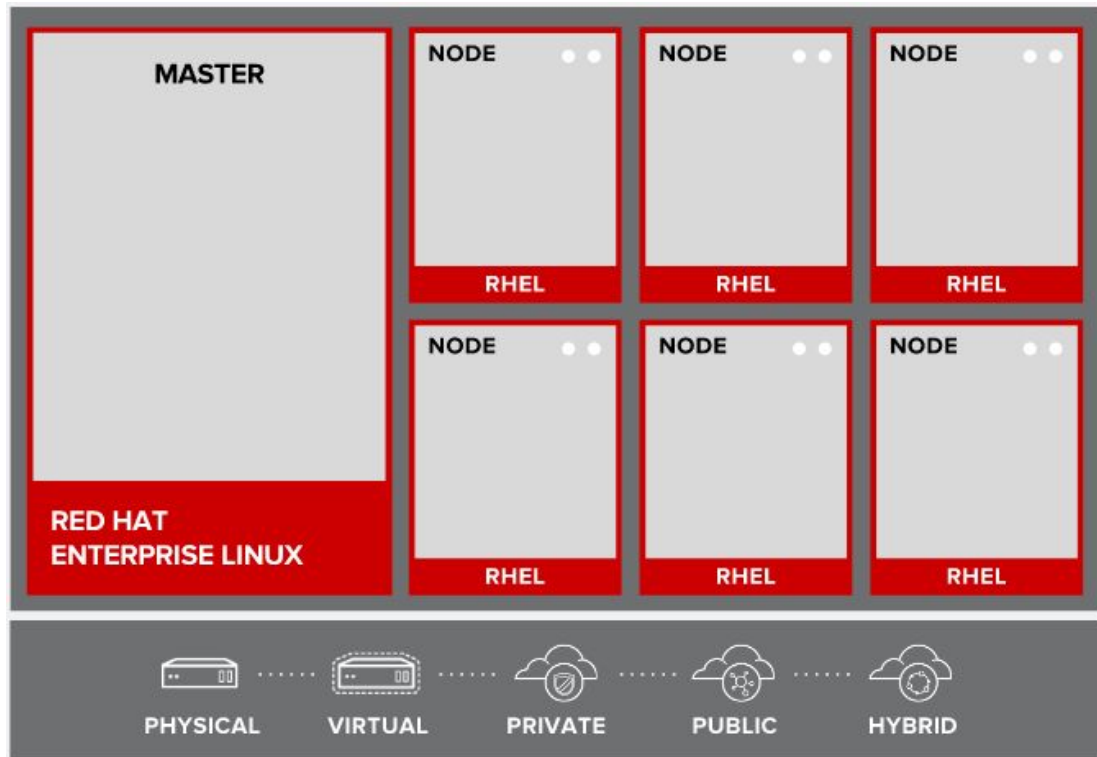


# PODS ARE THE UNIT OF ORCHESTRATION

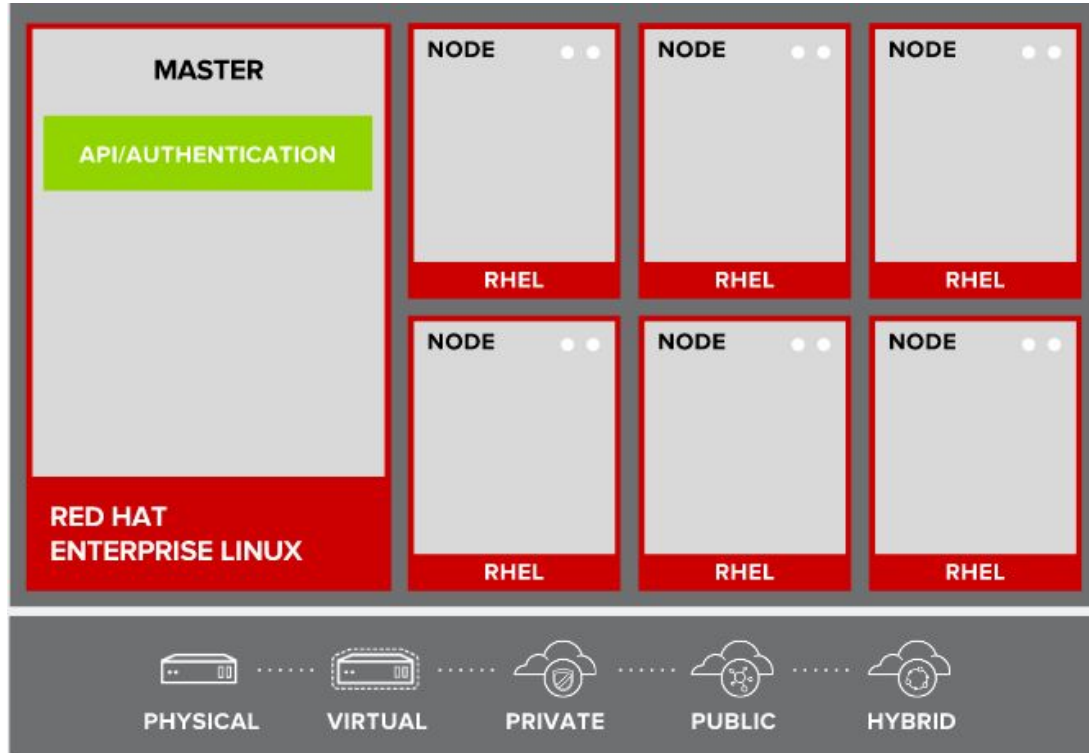




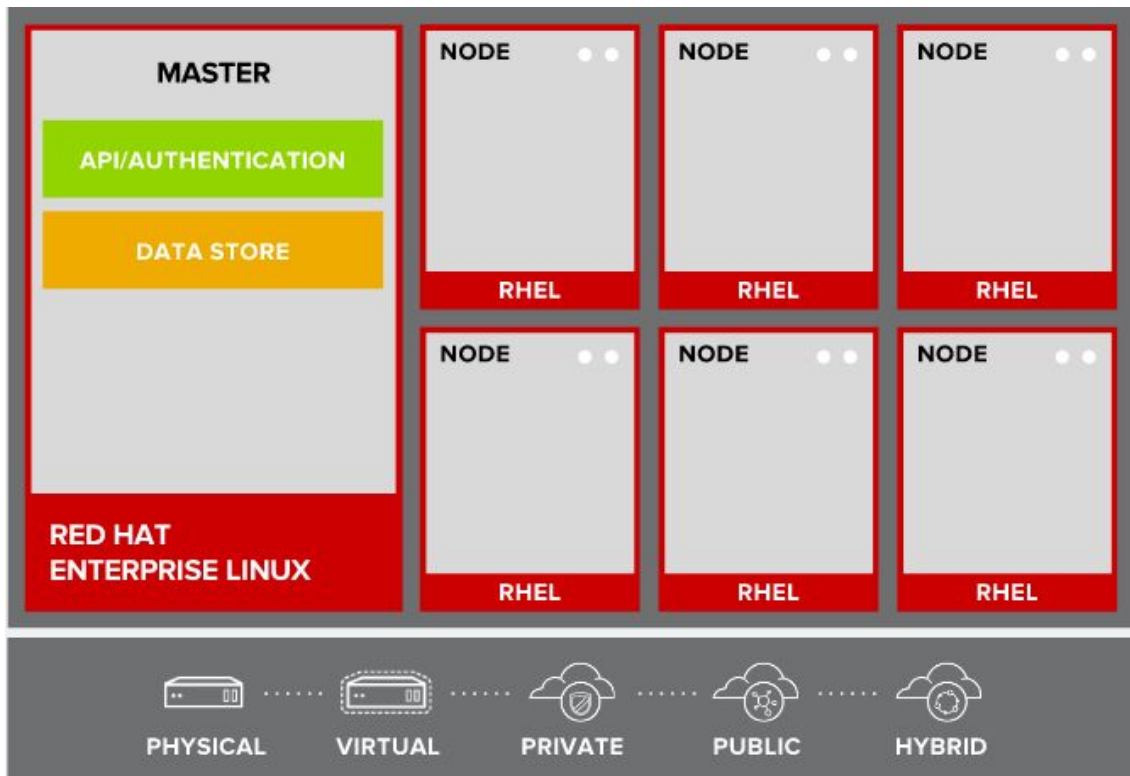
# MASTERS ARE THE CONTROL PLANE



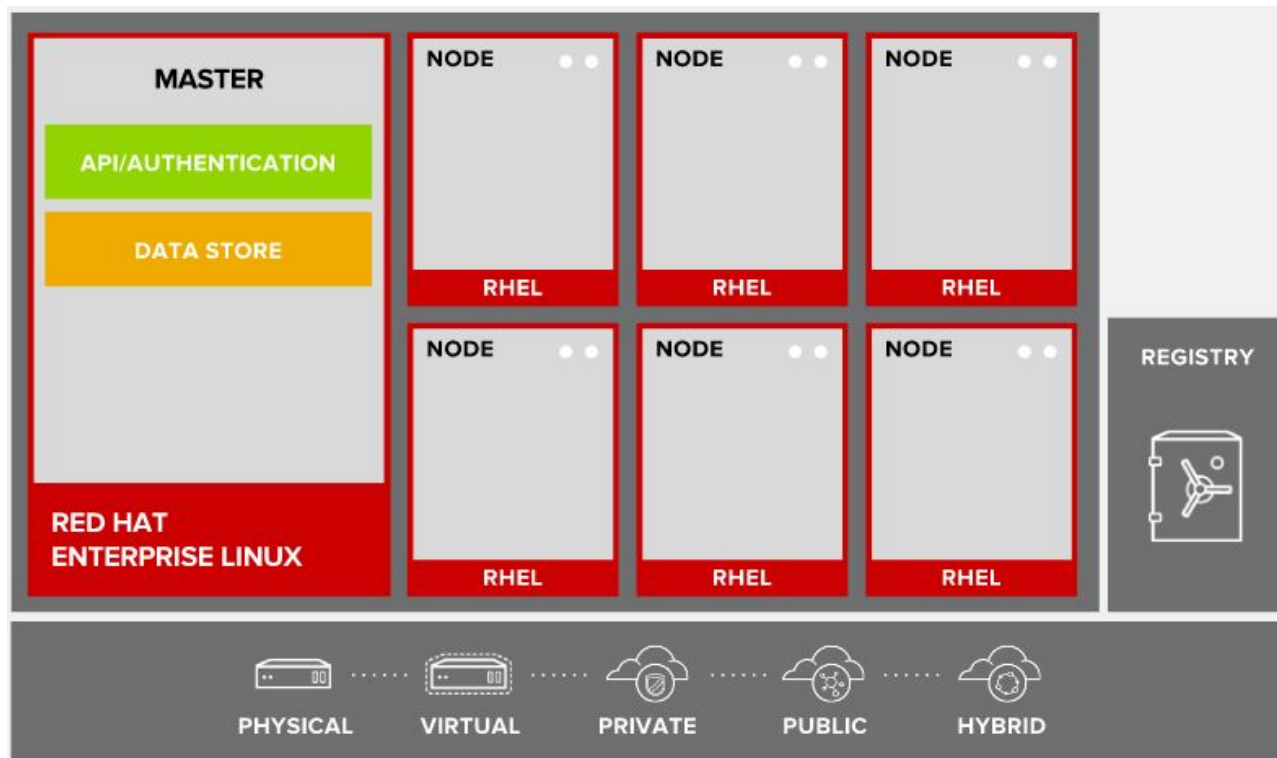
# API AND AUTHENTICATION



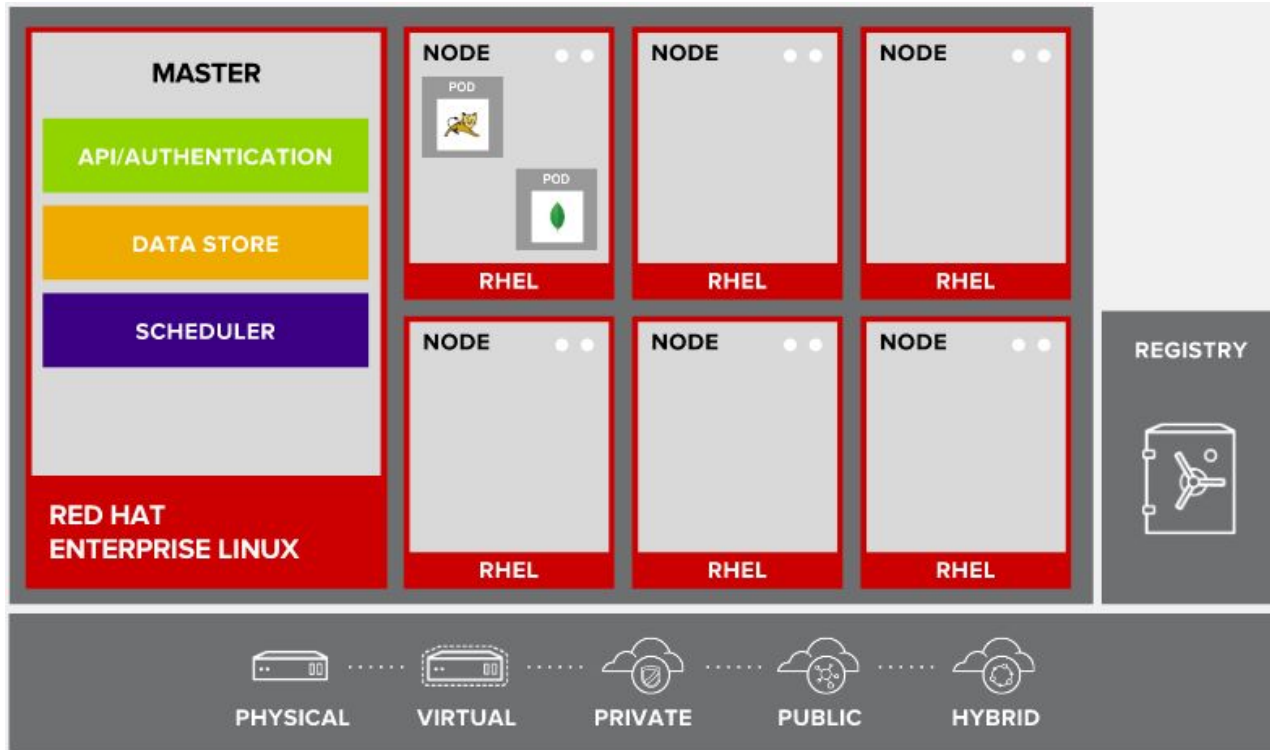
# DESIRED AND CURRENT STATE



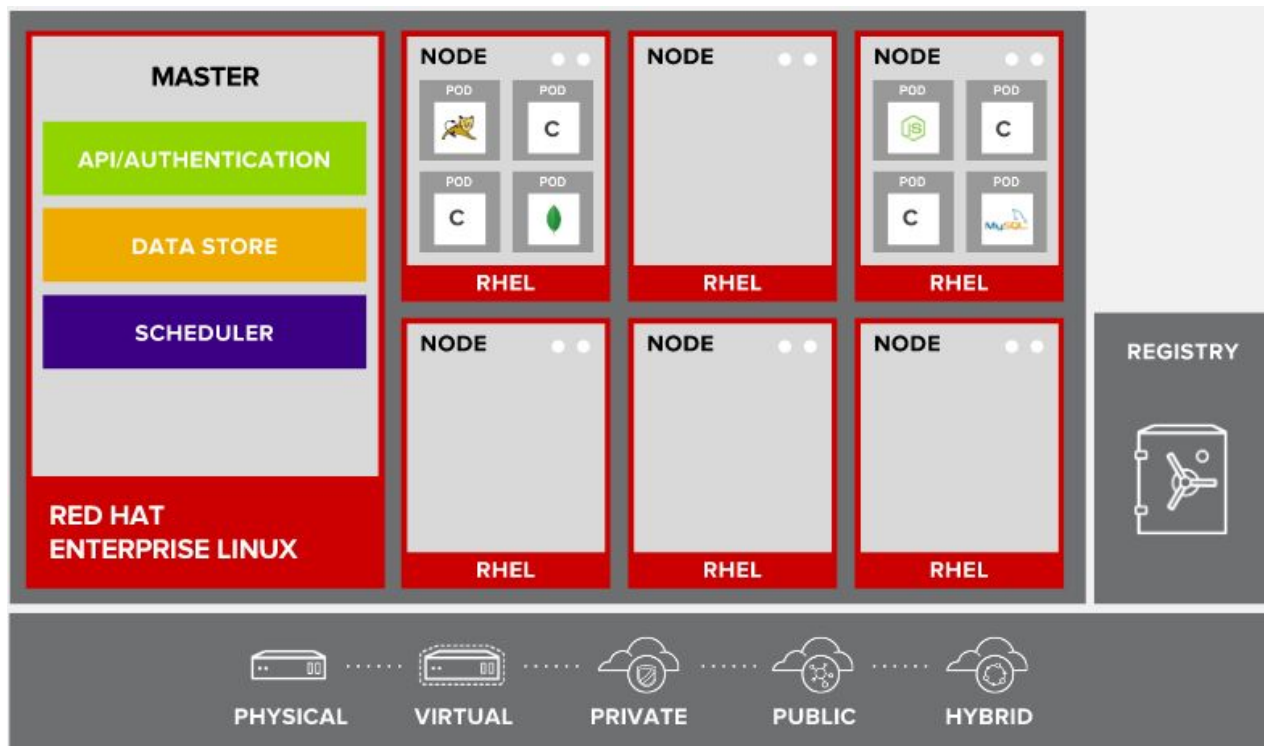
# INTEGRATED CONTAINER REGISTRY



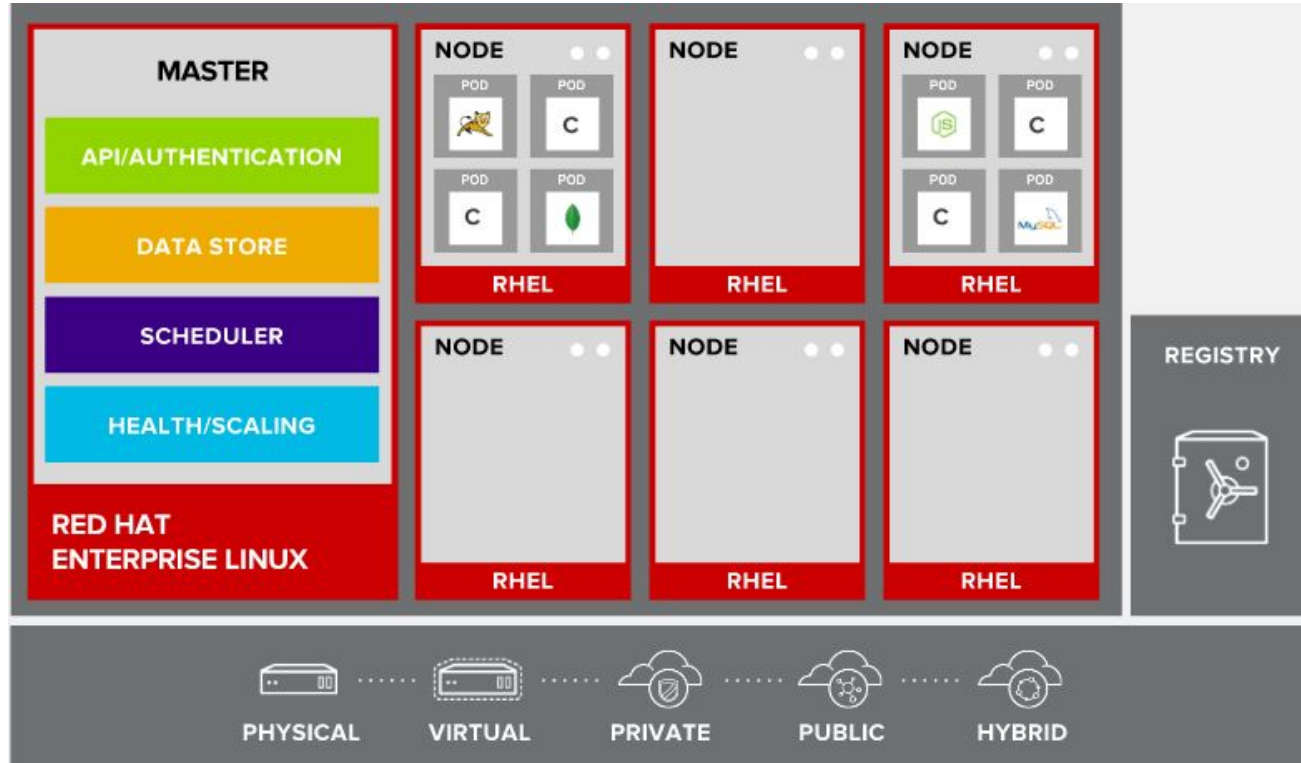
# ORCHESTRATION AND SCHEDULING



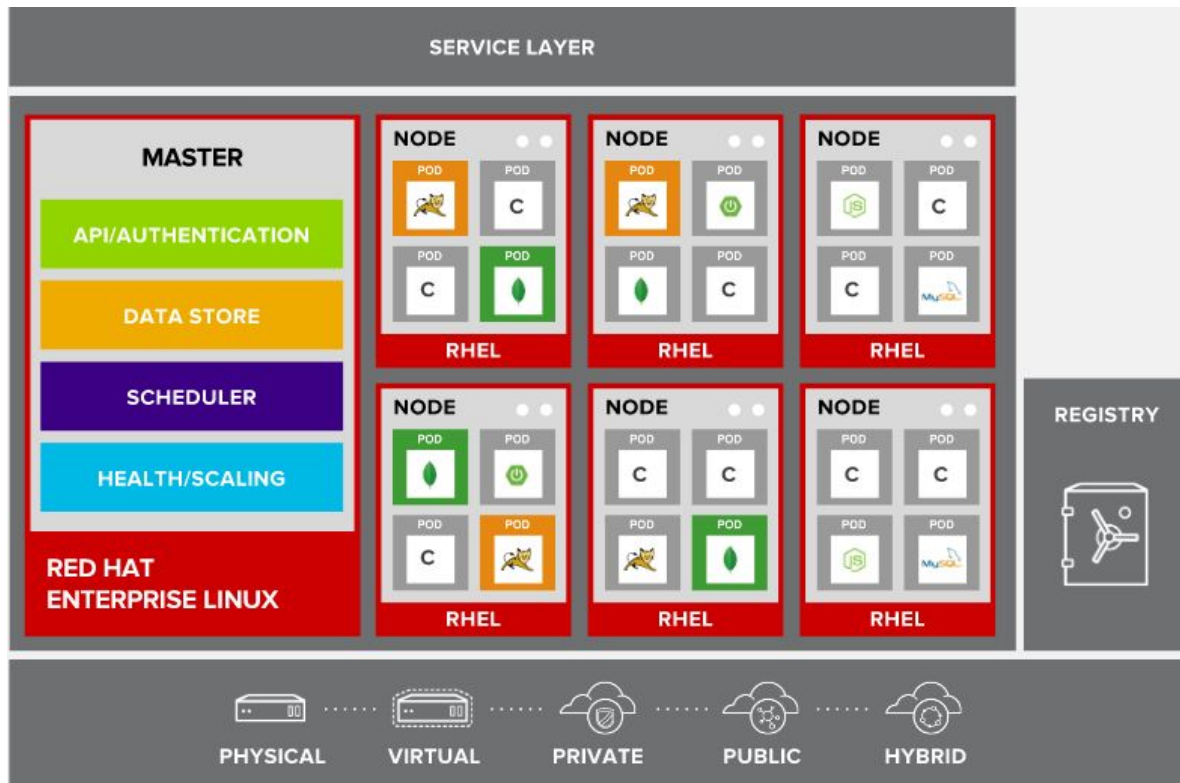
# PLACEMENT BY POLICY



# AUTOSCALING PODS

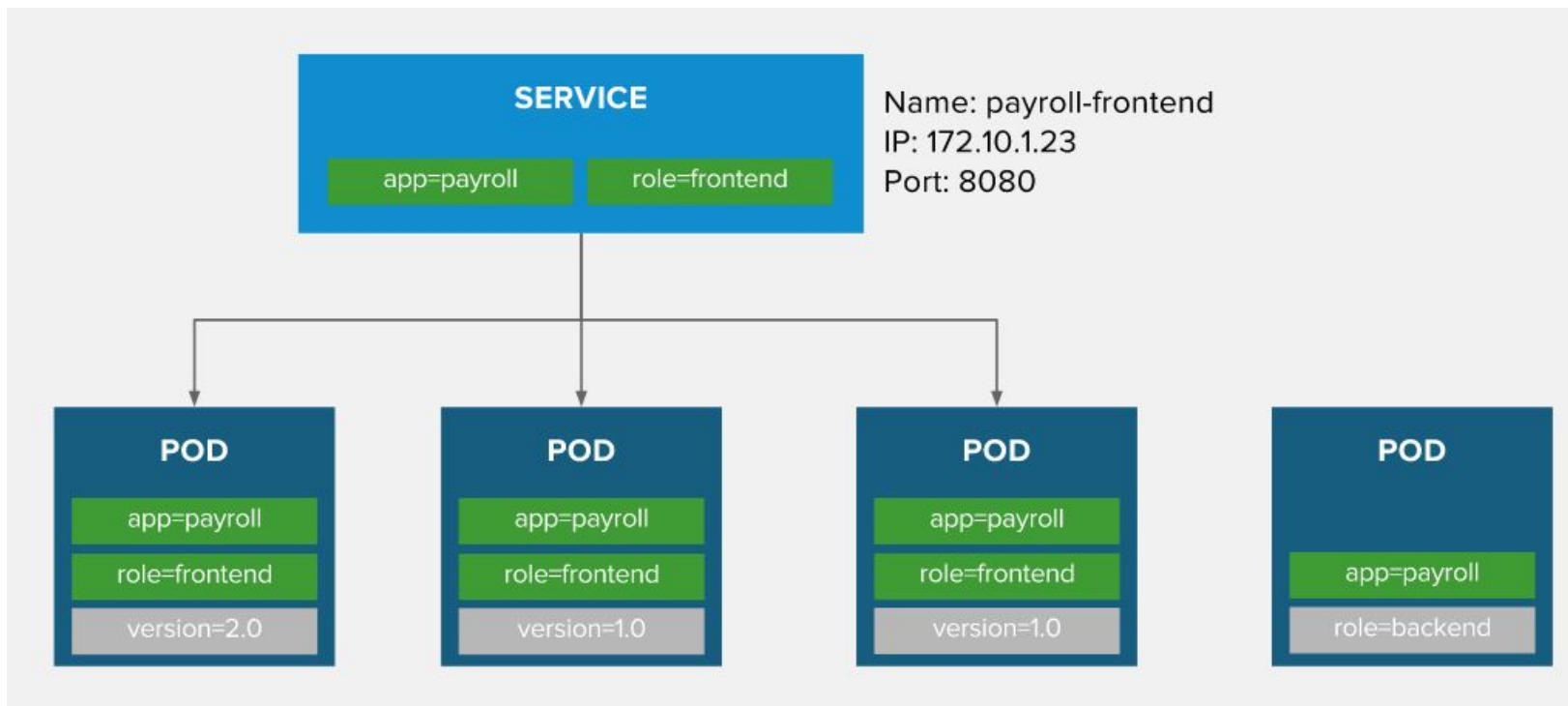


# SERVICE DISCOVERY

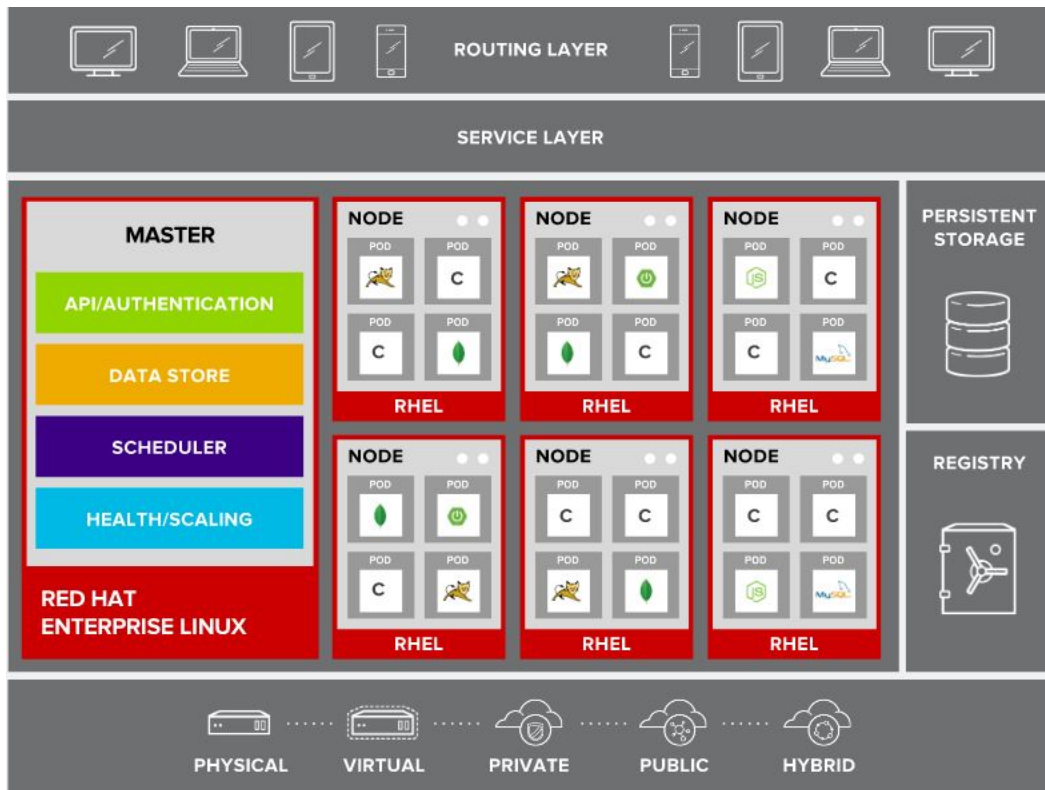




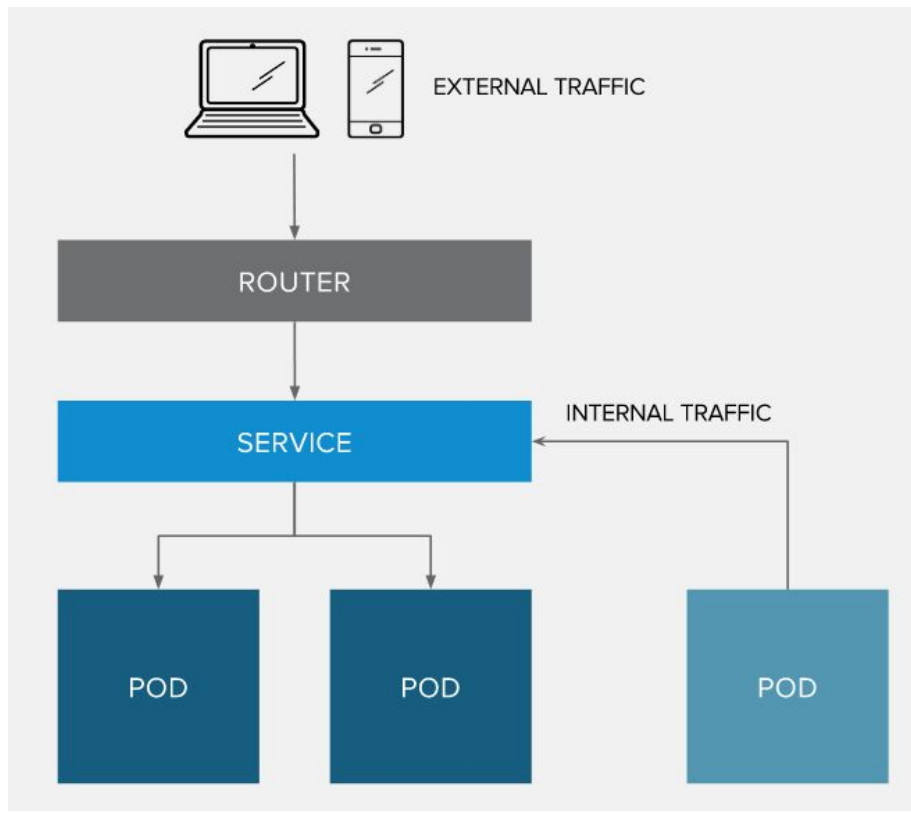
# BUILT-IN SERVICE DISCOVERY INTERNAL LOAD-BALANCING



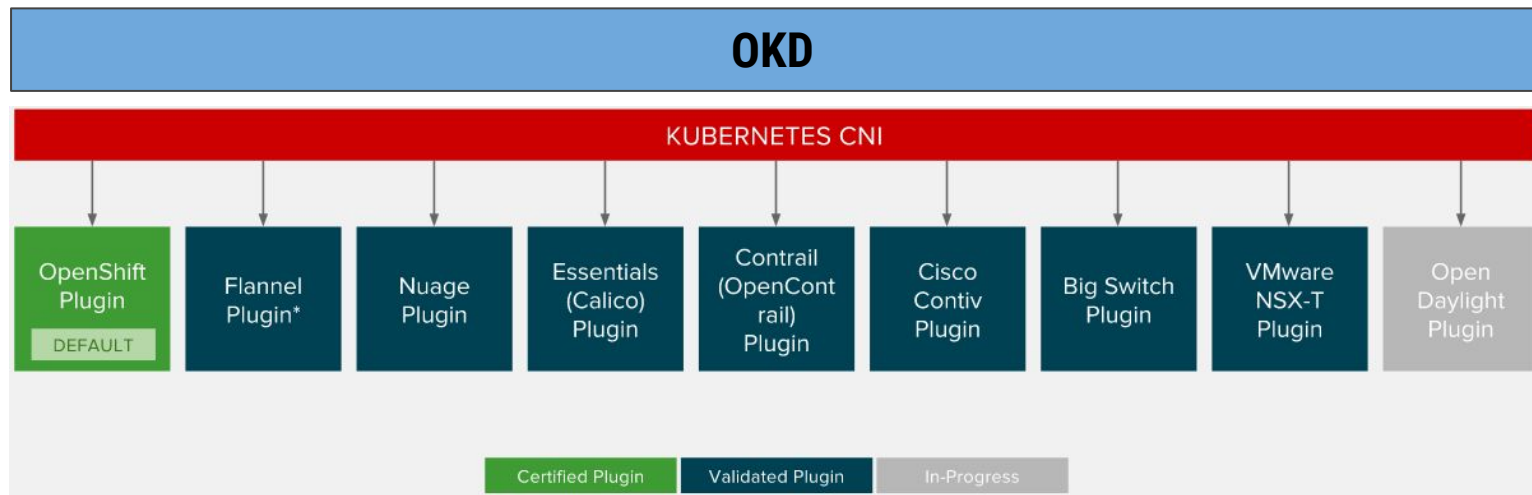
# ROUTING AND LOAD-BALANCING



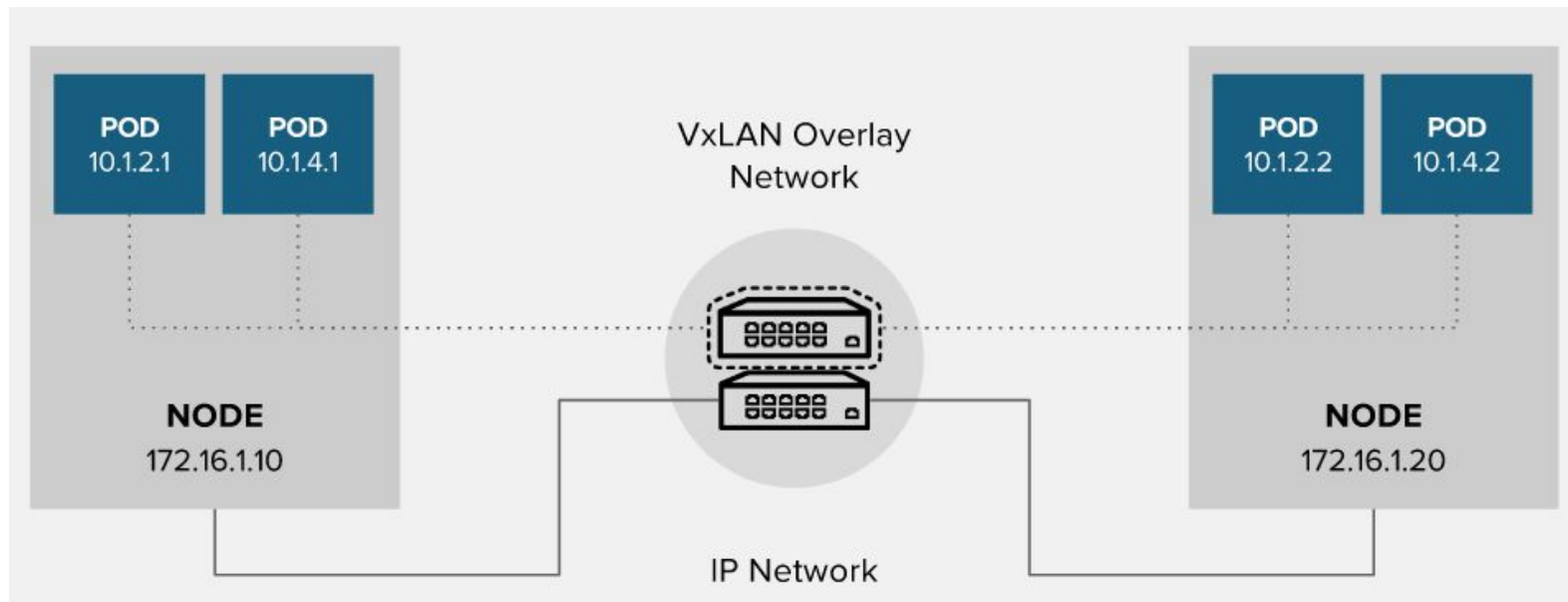
# ROUTE EXPOSES SERVICES EXTERNALLY



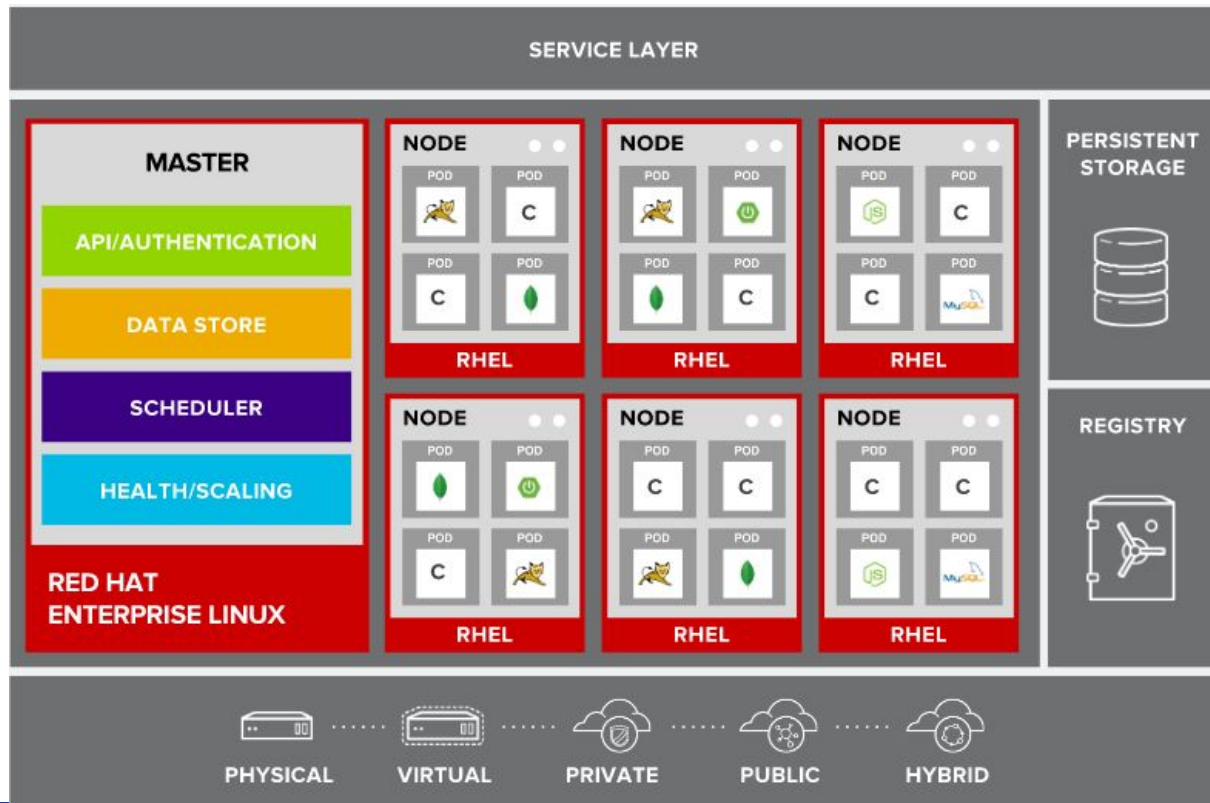
# OKD NETWORK PLUGINS



# OKD NETWORKING



# PERSISTENT DATA IN CONTAINERS

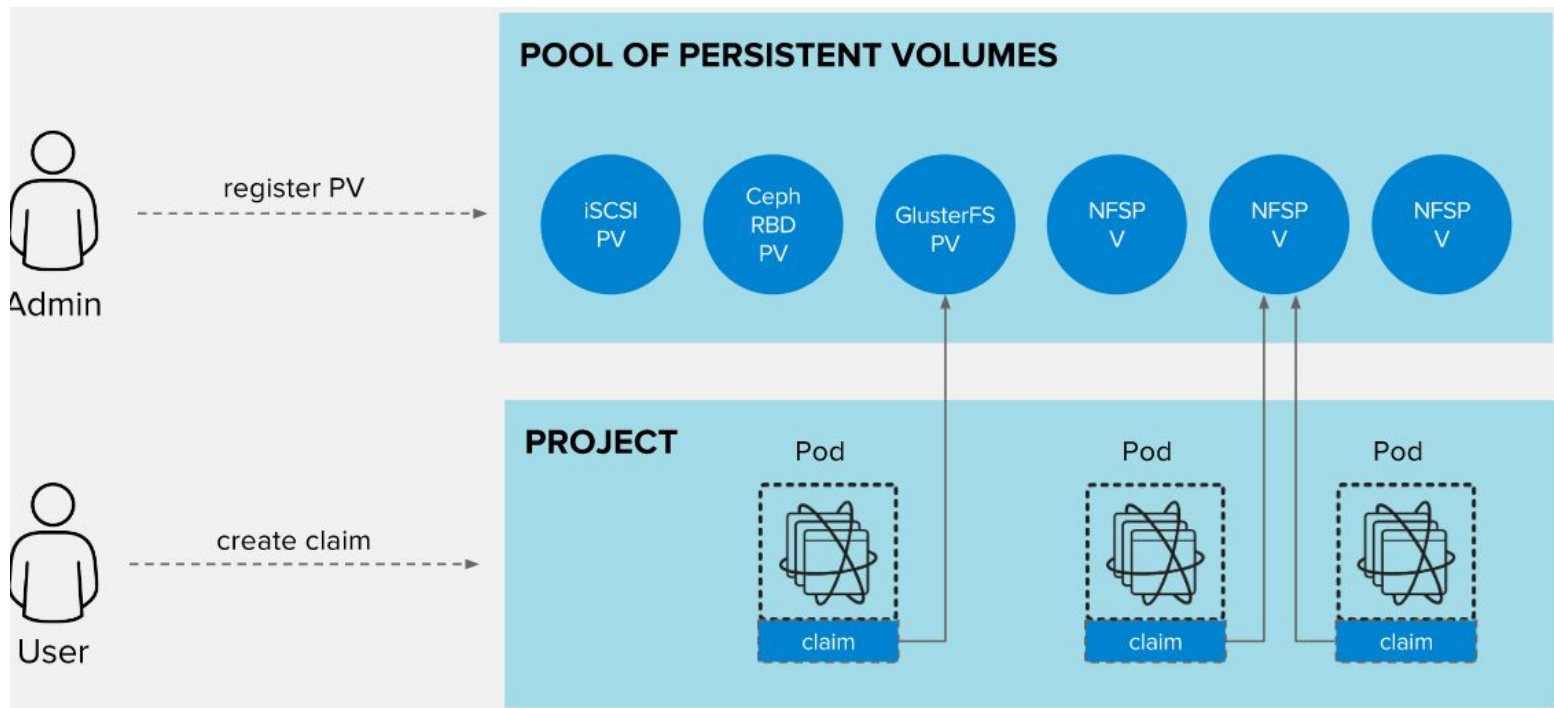


# PERSISTENT STORAGE

- 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

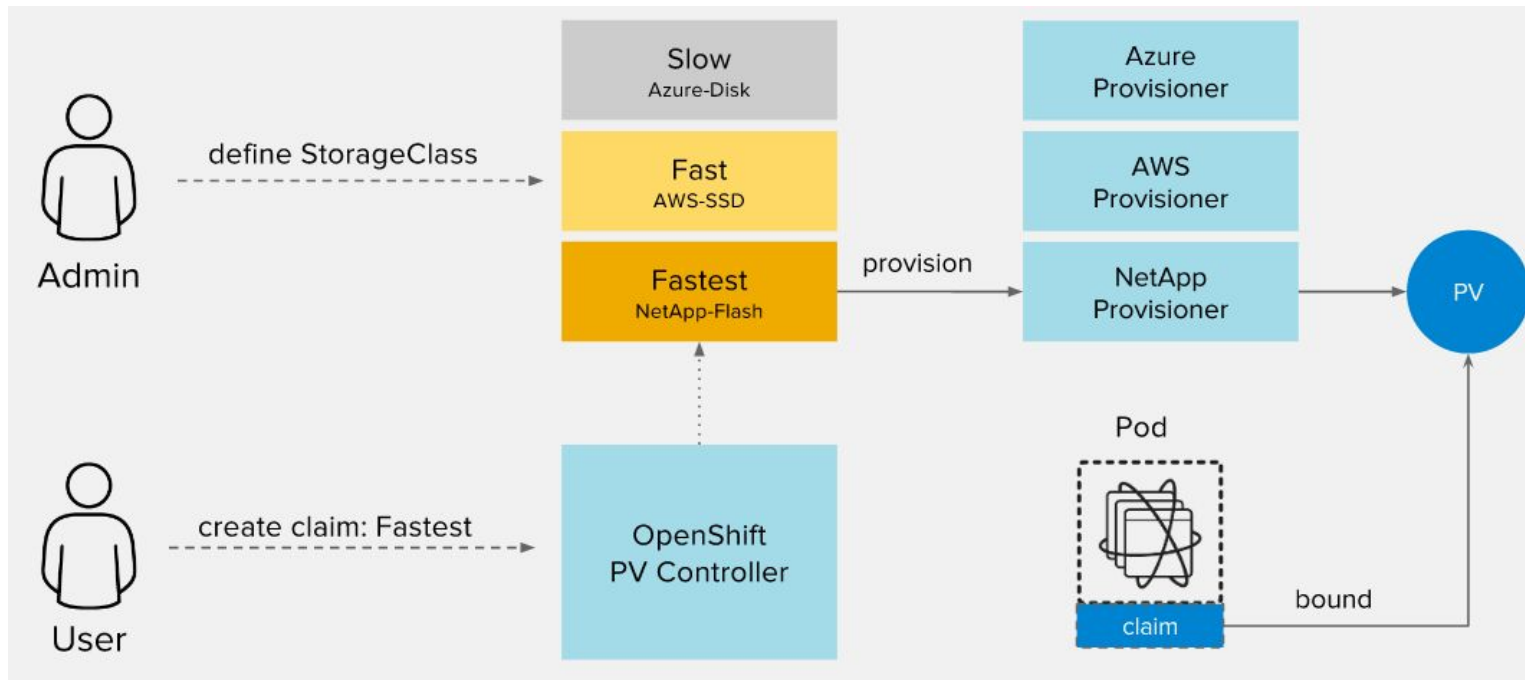
NFS	OpenStack Cinder	iSCSI	Azure Disk	AWS EBS	FlexVolume
GlusterFS	Ceph RBD	Fiber Channel	Azure File	GCE Persistent Disk	VMWare vSphere VMDK

# PERSISTENT STORAGE

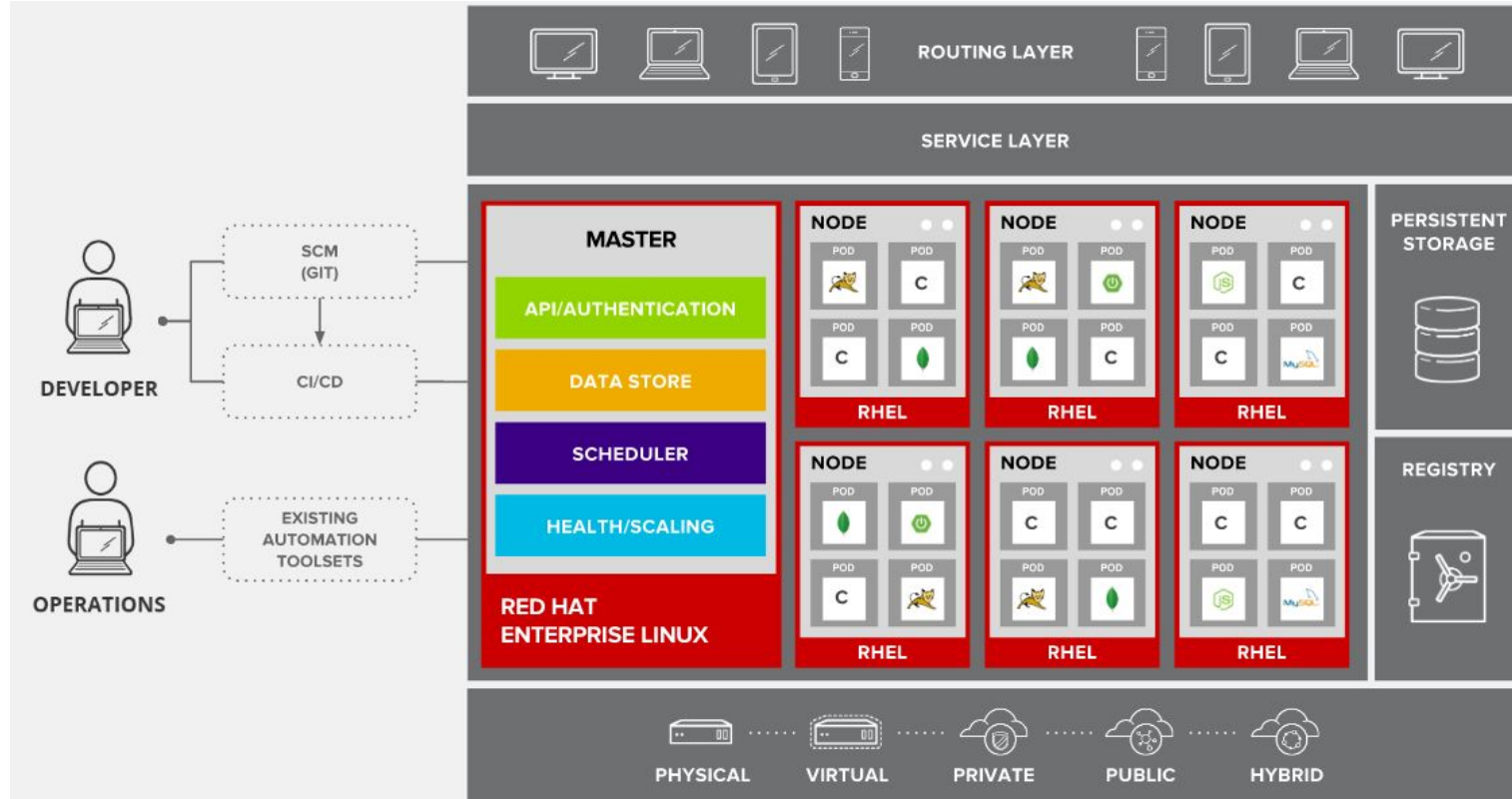




# DYNAMIC VOLUME PROVISIONING



# ACCESS VIA WEB, CLI, IDE AND API



# BUILD AND DEPLOY CONTAINER IMAGE



# BUILD AND DEPLOY CONTAINER IMAGES



**DEPLOY YOUR  
SOURCE CODE**



**DEPLOY YOUR  
APP BINARY**

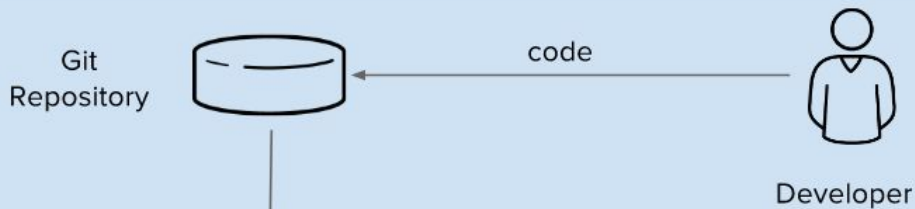


**DEPLOY YOUR  
CONTAINER IMAGE**

# DEPLOY SOURCE CODE WITH SOURCE-TO-IMAGE (S2I)

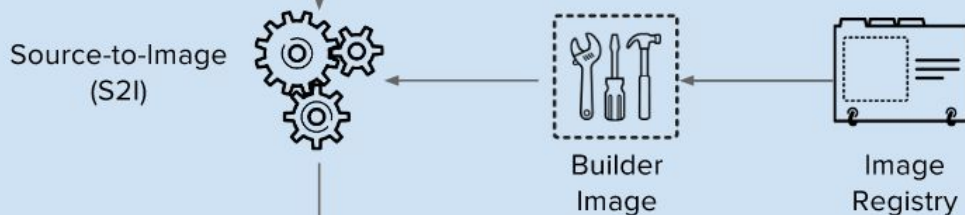
## BUILD APP (OpenShift)

(OpenShift)



## BUILD IMAGE (OpenShift)

(OpenShift)

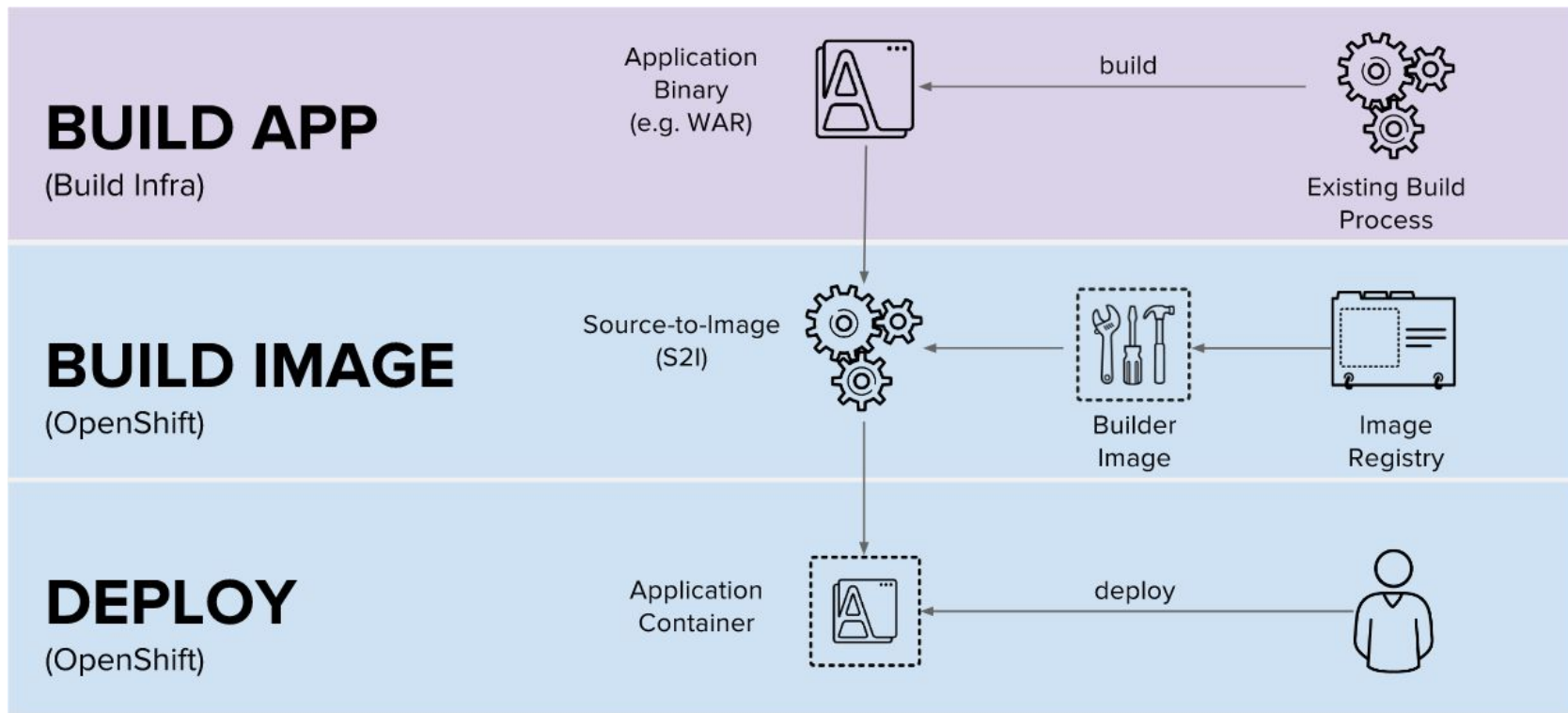


## DEPLOY (OpenShift)

(OpenShift)



# DEPLOY APP BINARY WITH SOURCE-TO-IMAGE (S2I)



# DEPLOY DOCKER IMAGE

## BUILD IMAGE

(Build Infra)

Application Image



build

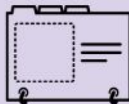


Existing Image  
Build Process

## PUSH

(Build Infra)

Image  
Registry



## DEPLOY

(Openshift)

Application  
Container



deploy



Demo Time!





# Q&A



<https://www.okd.io/>

<https://github.com/alanadiprastyo/meetup-okd-001-JKT>