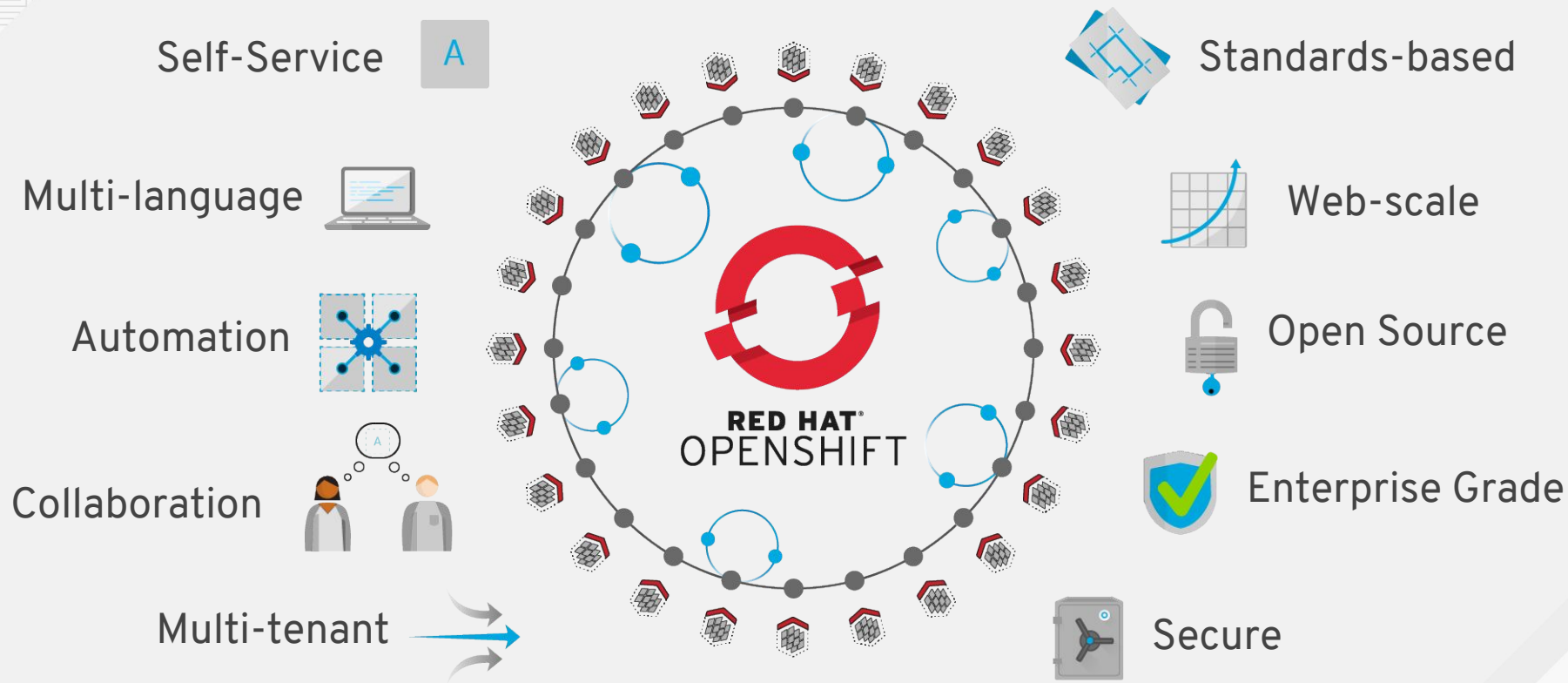
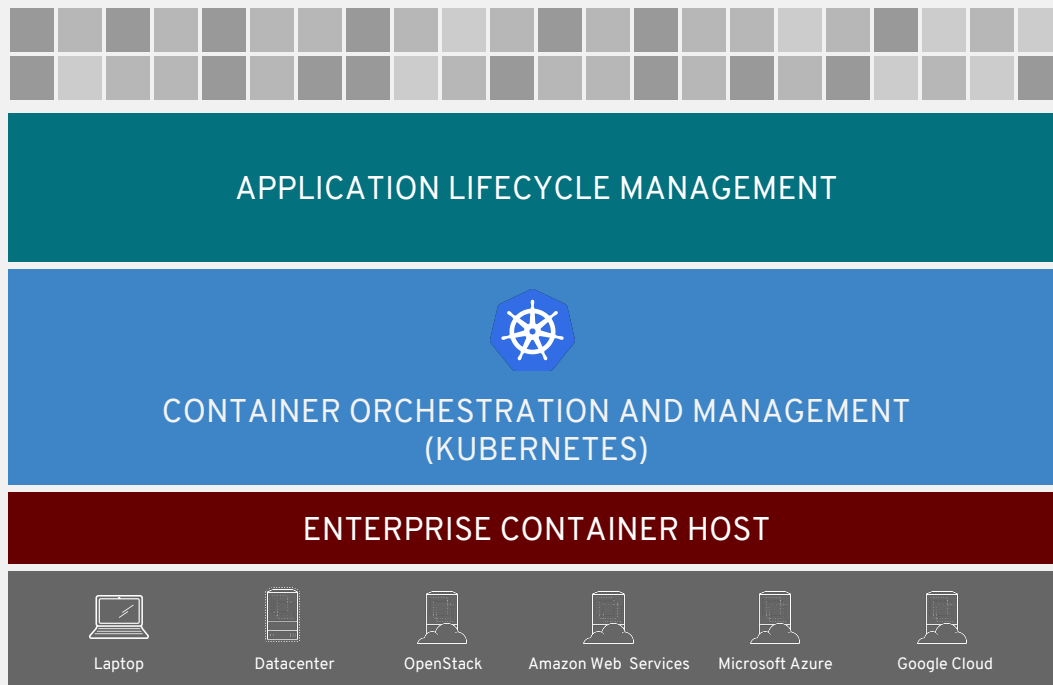




OPENSIFT CONTAINER PLATFORM TECHNICAL OVERVIEW

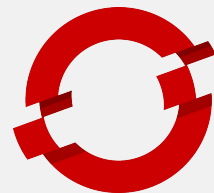


OPENSIFT CONTAINER PLATFORM



ANY
CONTAINER

.....

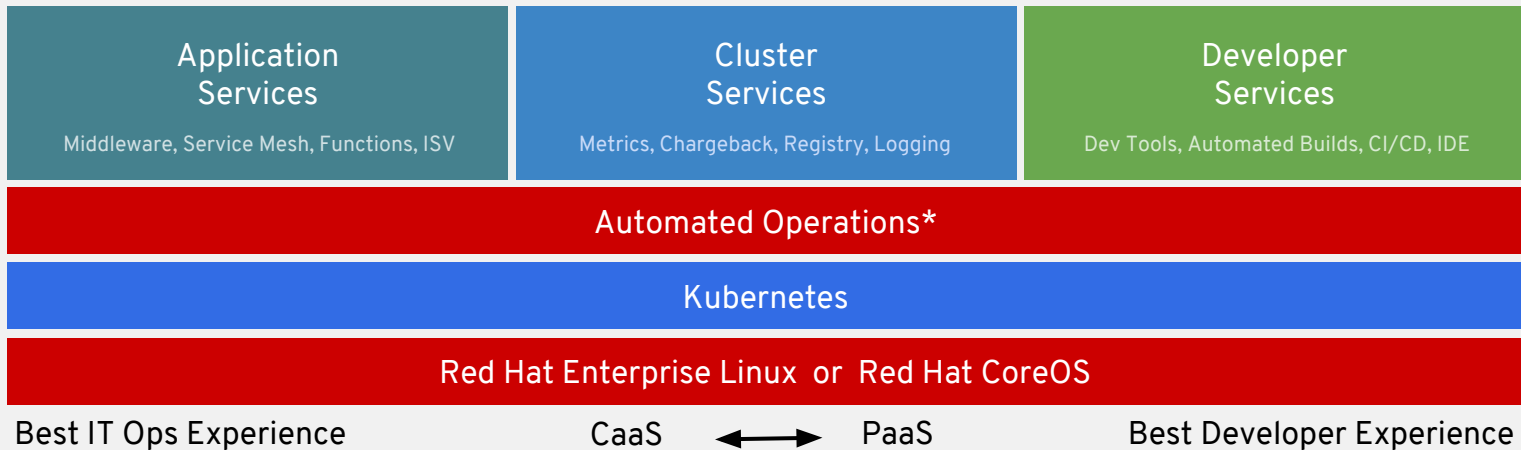


RED HAT
OPENSIFT

.....

ANY
INFRASTRUCTURE

OPENSIFT CONTAINER PLATFORM



*coming soon

LINUX CONTAINERS

WHAT ARE CONTAINERS?

It Depends Who You Ask



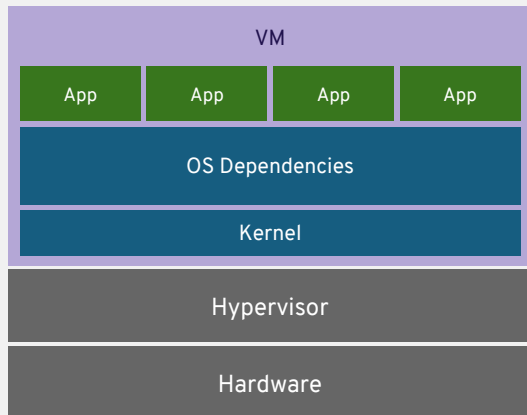
INFRASTRUCTURE

APPLICATIONS

- Application processes on a shared kernel
- Simpler, lighter, and denser than VMs
- Portable across different environments
- Package apps with all dependencies
- Deploy to any environment in seconds
- Easily accessed and shared

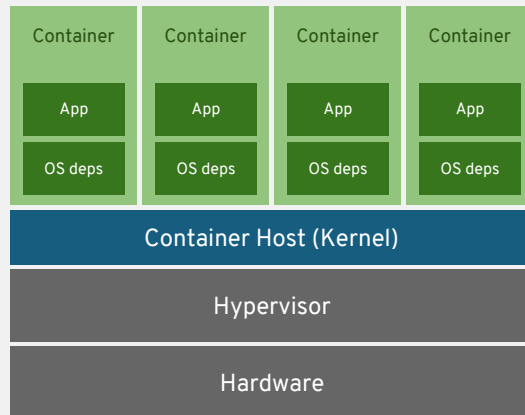
VIRTUAL MACHINES AND CONTAINERS

VIRTUAL MACHINES



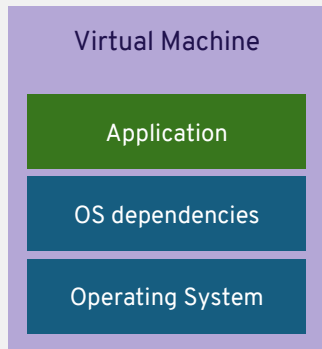
VM isolates the hardware

CONTAINERS

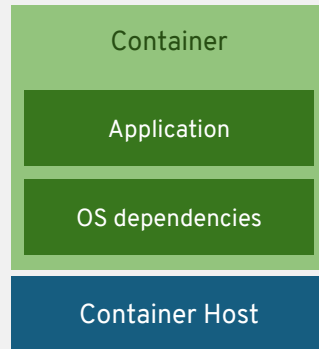


Container isolates the process

VIRTUAL MACHINES AND CONTAINERS



- + VM Isolation
- Complete OS
- Static Compute
- Static Memory
- High Resource Usage



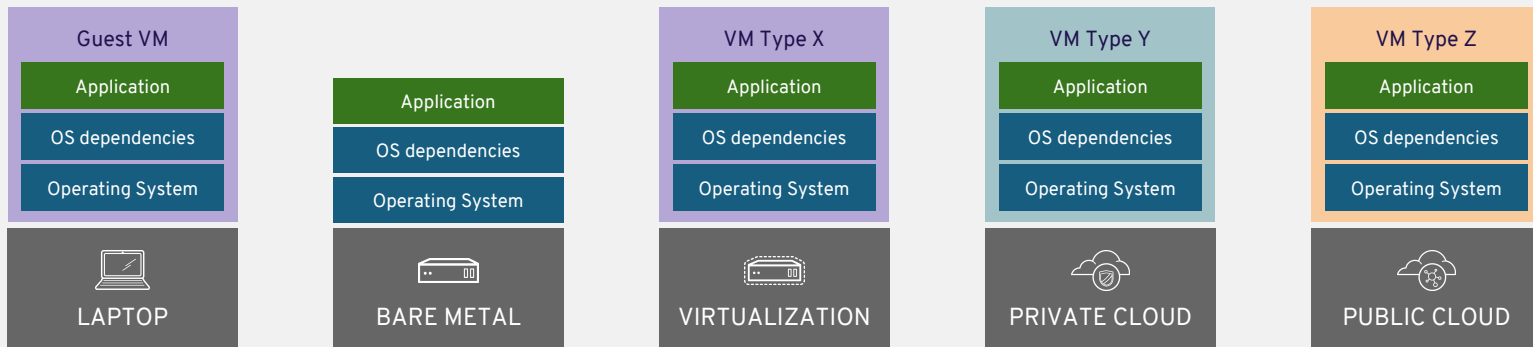
- + Container Isolation
- + Shared Kernel
- + Burstable Compute
- + Burstable Memory
- + Low Resource Usage

VIRTUAL MACHINES AND CONTAINERS



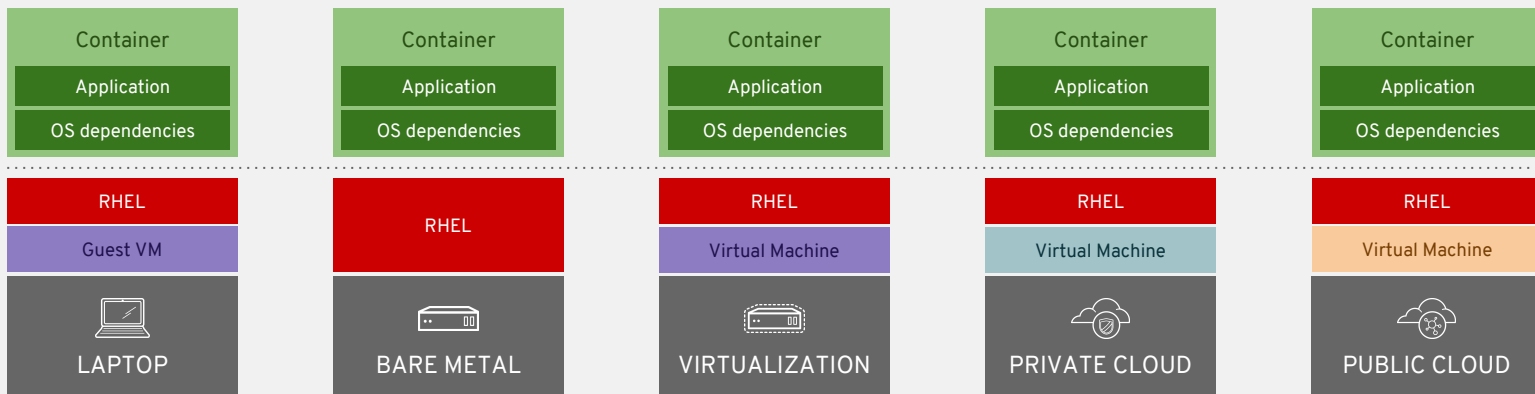
APPLICATION PORTABILITY WITH VM

Virtual machines are **NOT** portable across hypervisor and do **NOT** provide portable packaging for applications

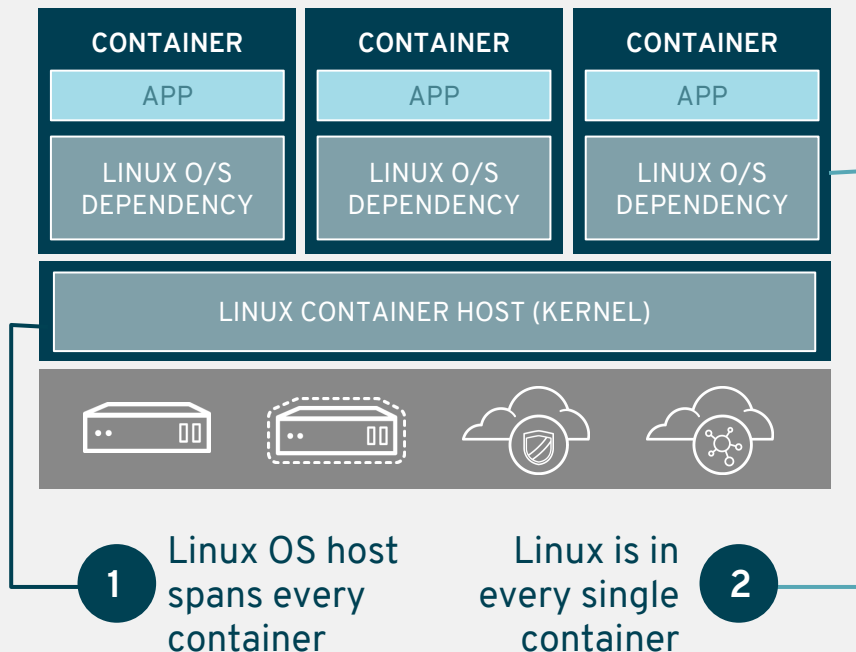


APPLICATION PORTABILITY WITH CONTAINERS

RHEL Containers + RHEL Host = Guaranteed Portability
Across Any Infrastructure



LINUX AND CONTAINER INFRASTRUCTURE

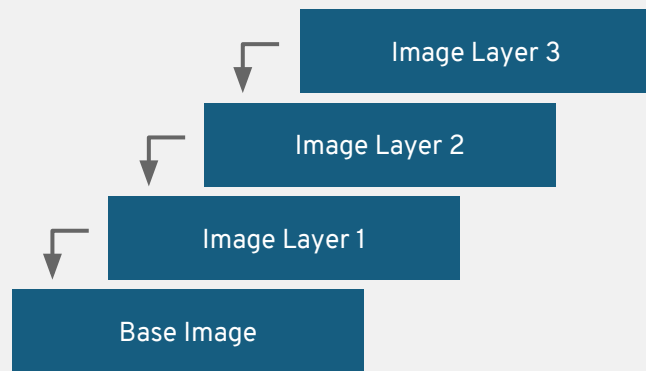


CONTAINERS ARE LINUX

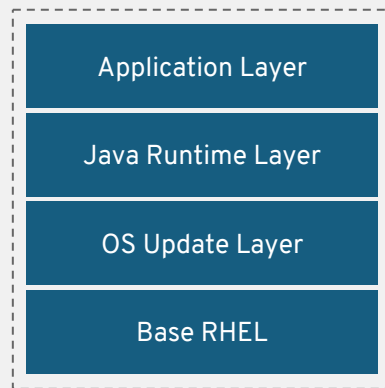
Red Hat
Enterprise Linux
is a leader in paid
Linux

70%
CY2016 paid
Linux share

RAPID SECURITY PATCHING USING CONTAINER IMAGE LAYERING



Container Image Layers



Example Container Image



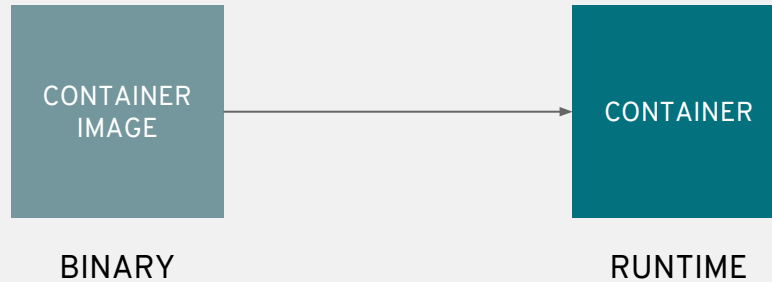
OPENSIFT CONCEPTS OVERVIEW

A container is the smallest compute unit

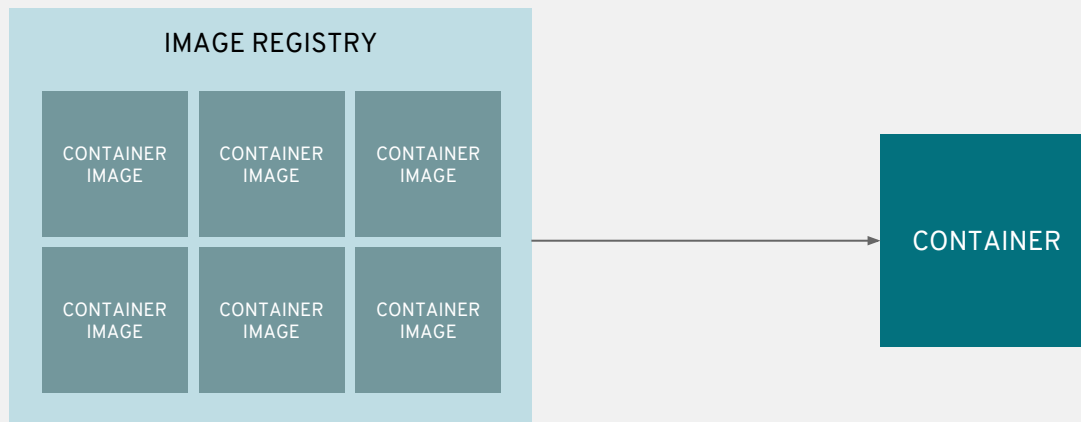


CONTAINER

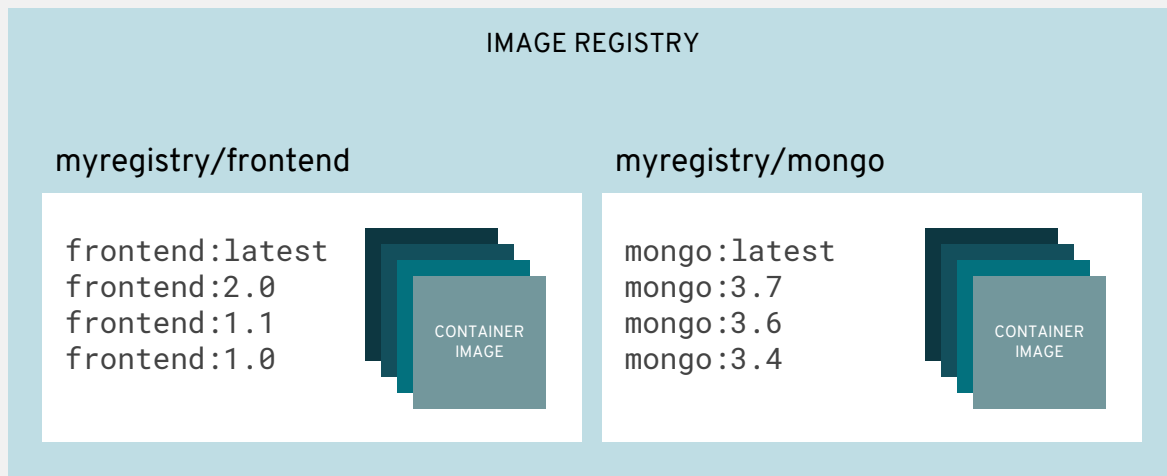
containers are created from container images



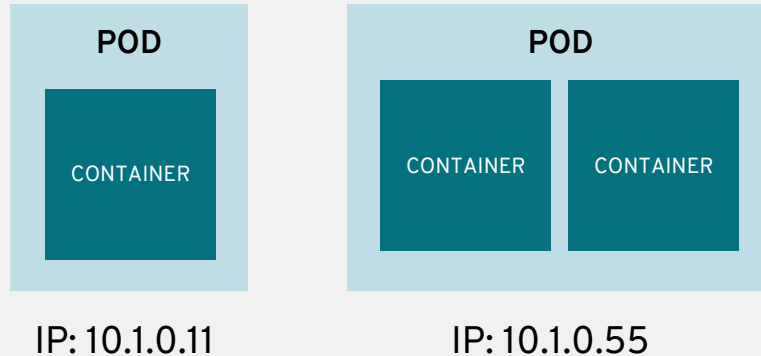
container images are stored in an image registry



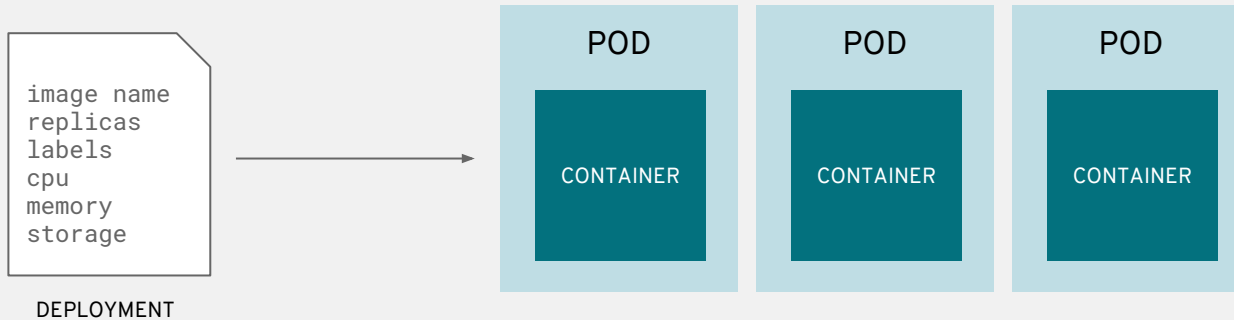
an image repository contains all versions of an image in the image registry



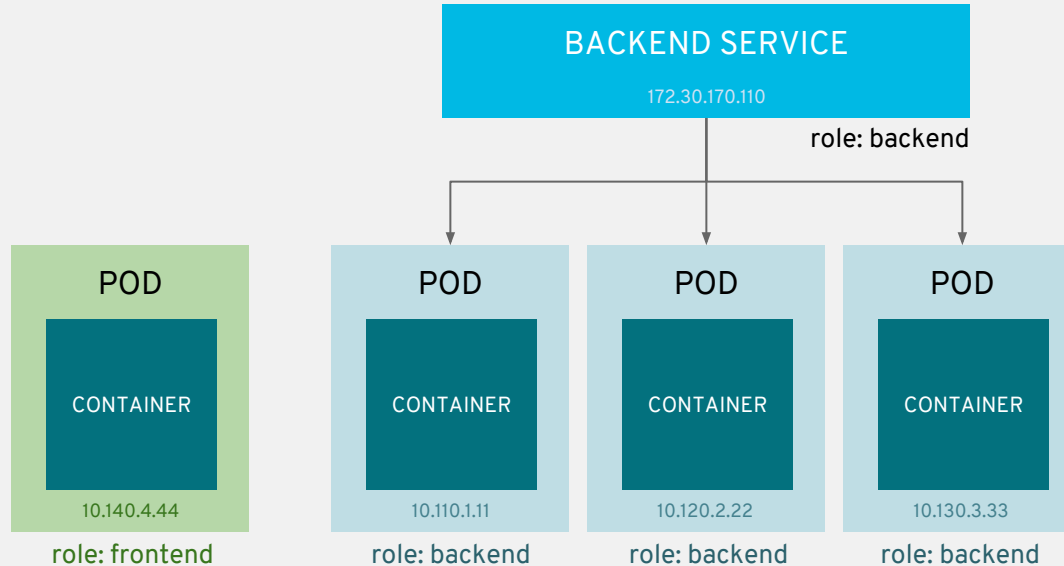
containers are wrapped in pods which are units of deployment and management



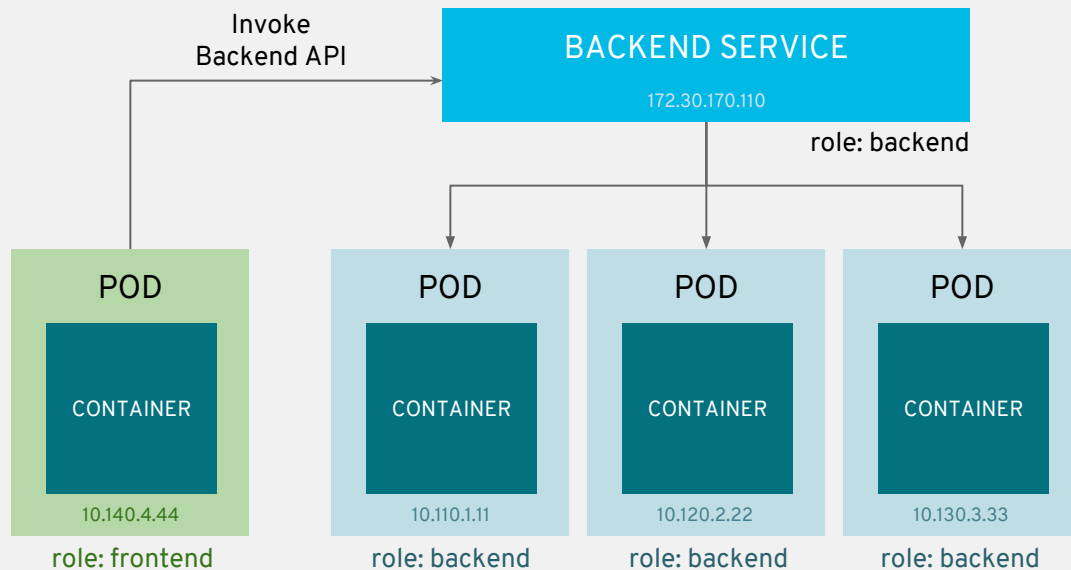
Pods configuration is defined in a deployment



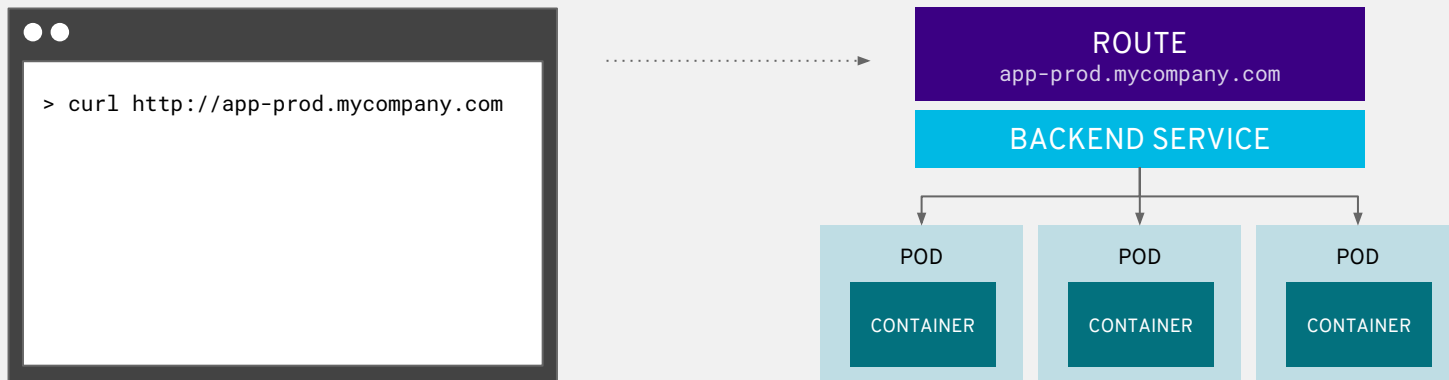
services provide internal load-balancing and service discovery across pods



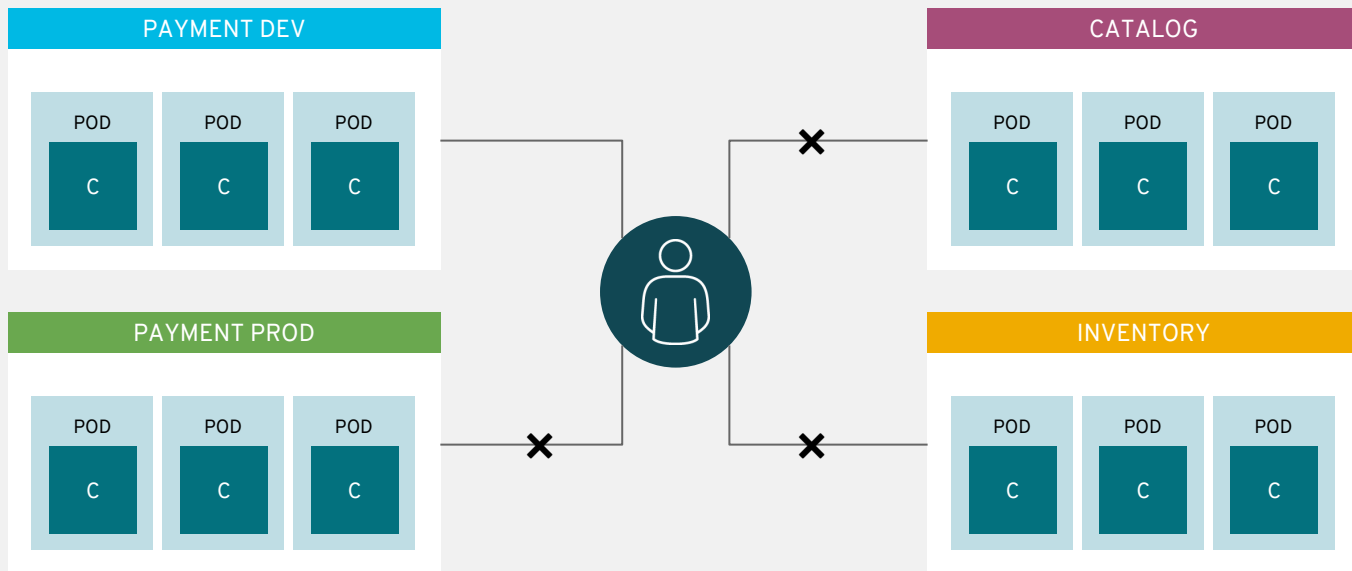
apps can talk to each other via services



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



projects isolate apps across environments,
teams, groups and departments

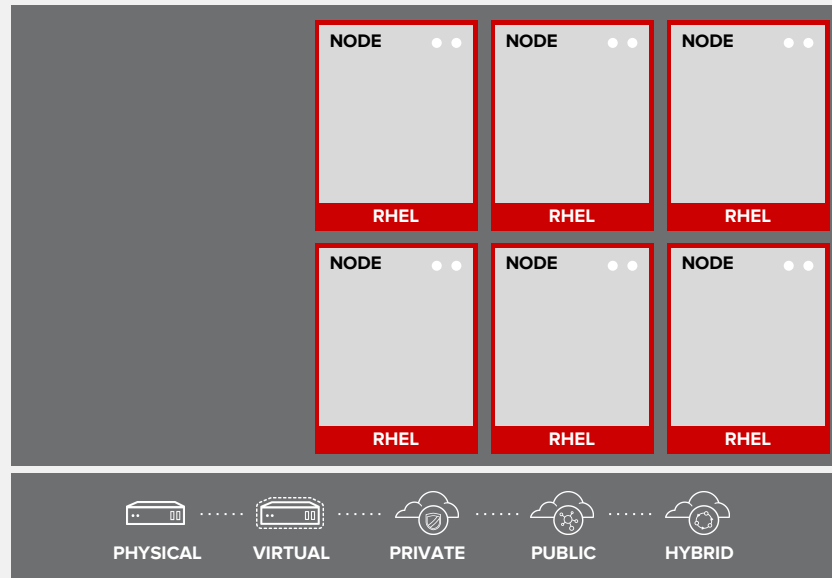


OPENSIFT ARCHITECTURE

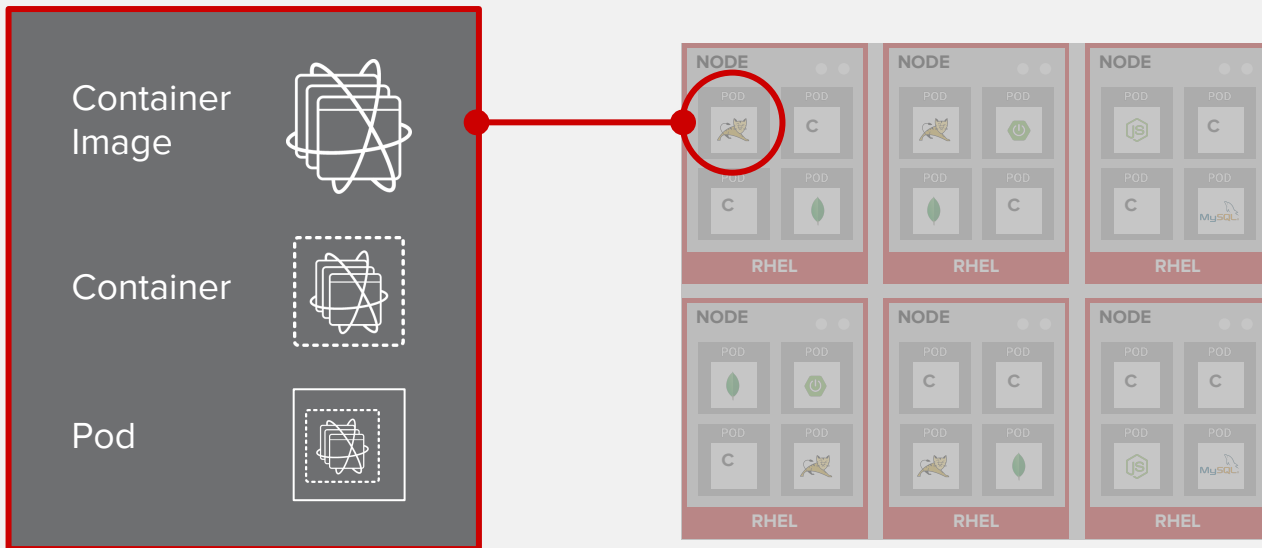
YOUR CHOICE OF INFRASTRUCTURE



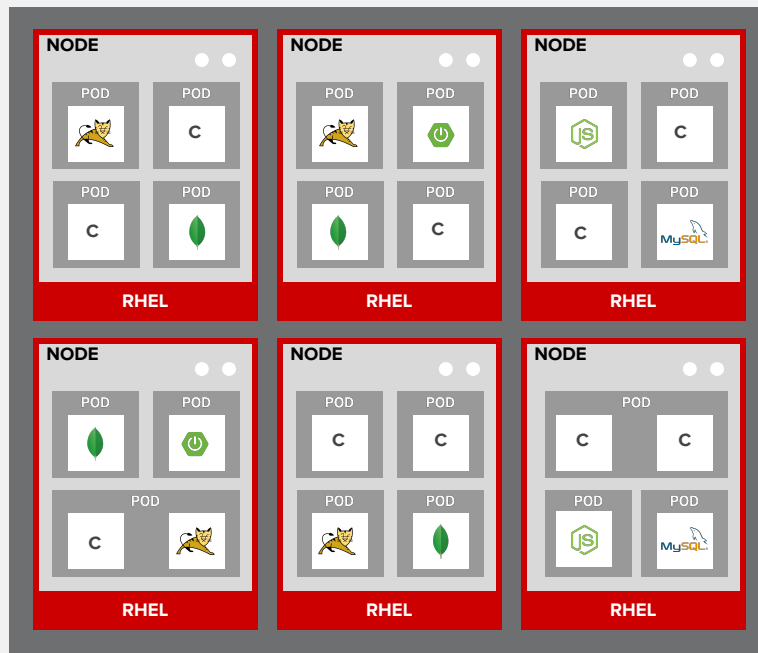
NODES RHEL INSTANCES WHERE APPS RUN



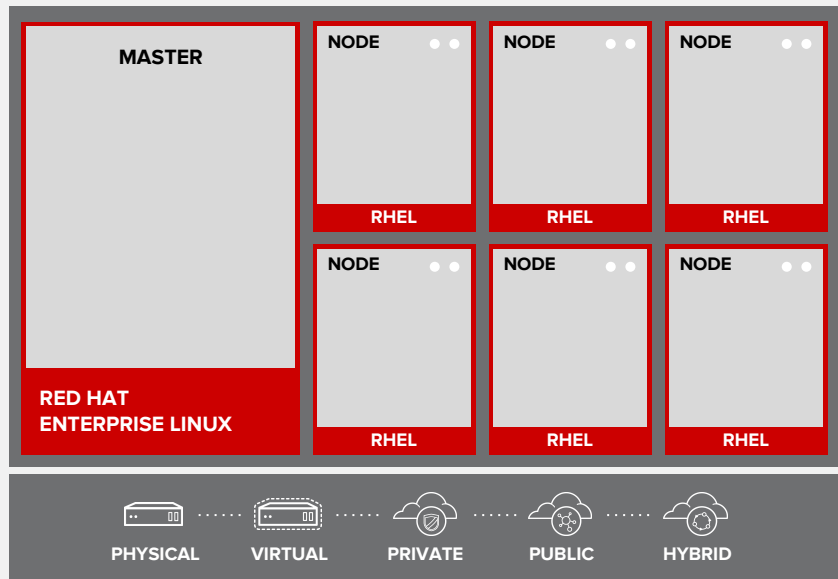
APPS RUN IN CONTAINERS



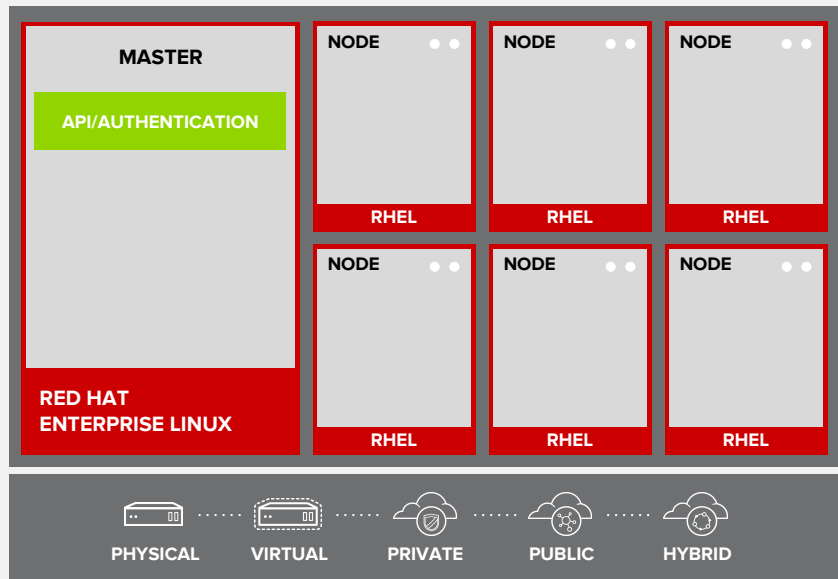
PODS ARE THE UNIT OF ORCHESTRATION



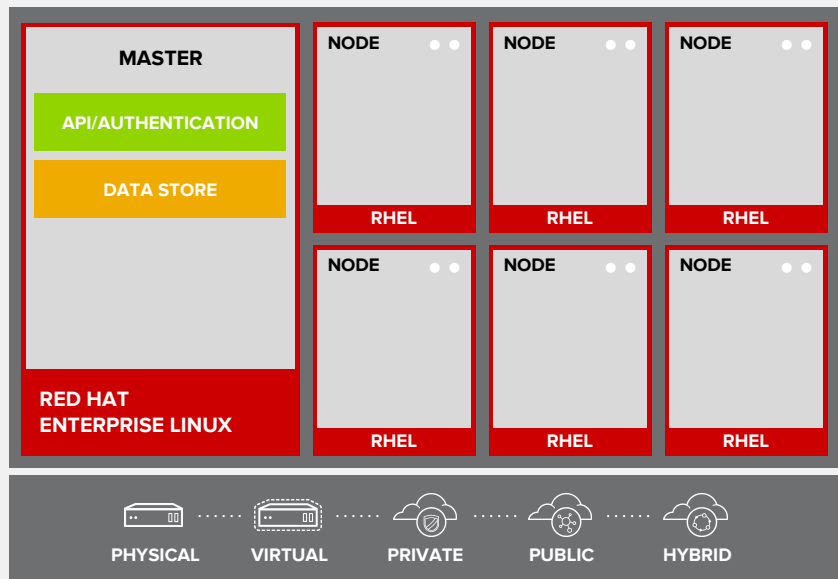
MASTERS ARE THE CONTROL PLANE



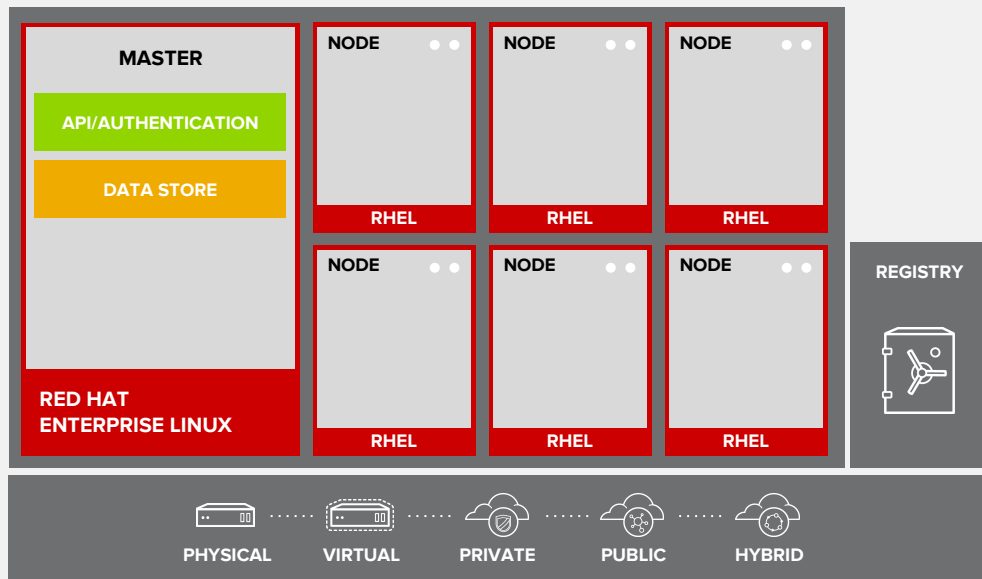
API AND AUTHENTICATION



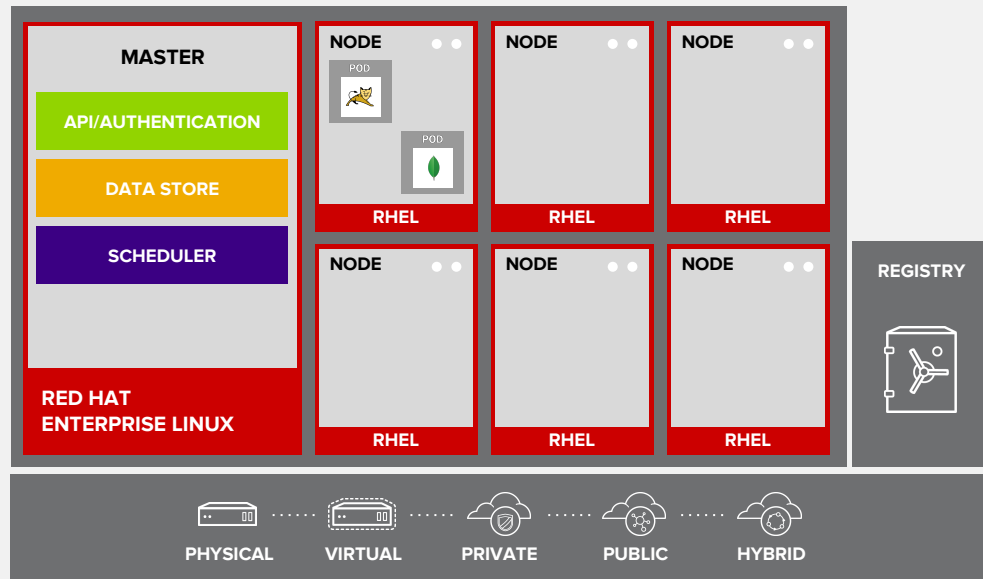
DESIRED AND CURRENT STATE



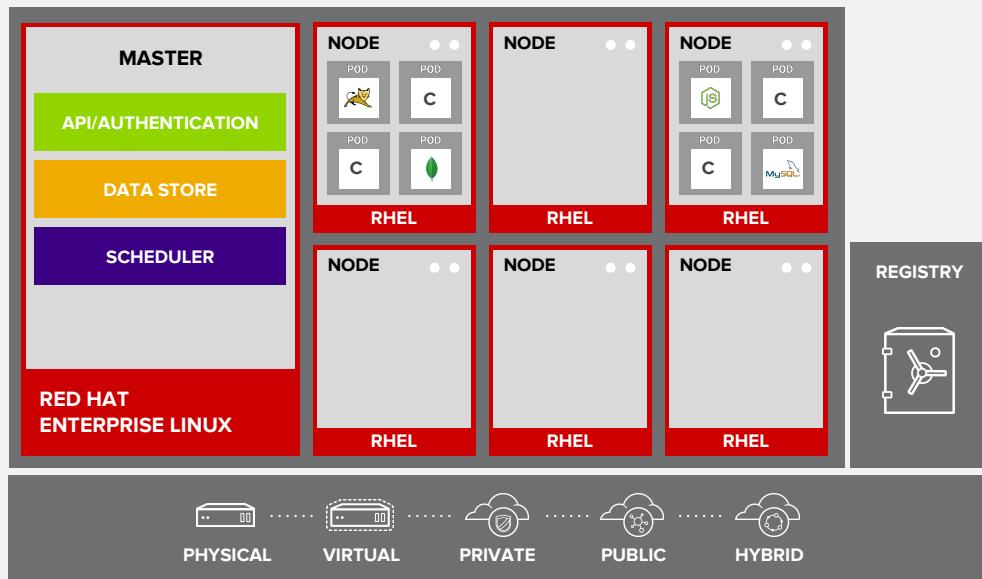
INTEGRATED CONTAINER REGISTRY



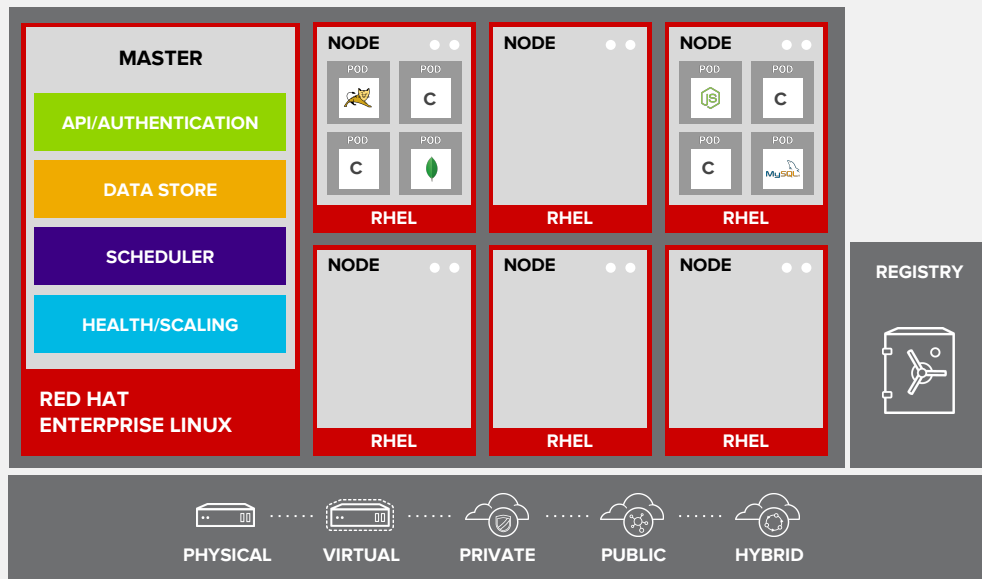
ORCHESTRATION AND SCHEDULING



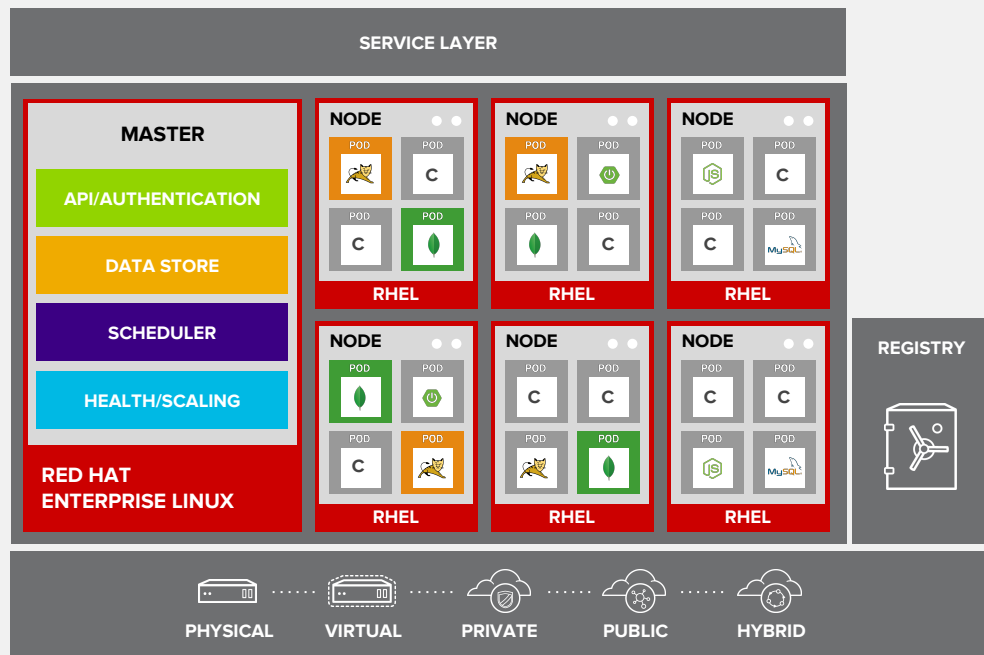
PLACEMENT BY POLICY



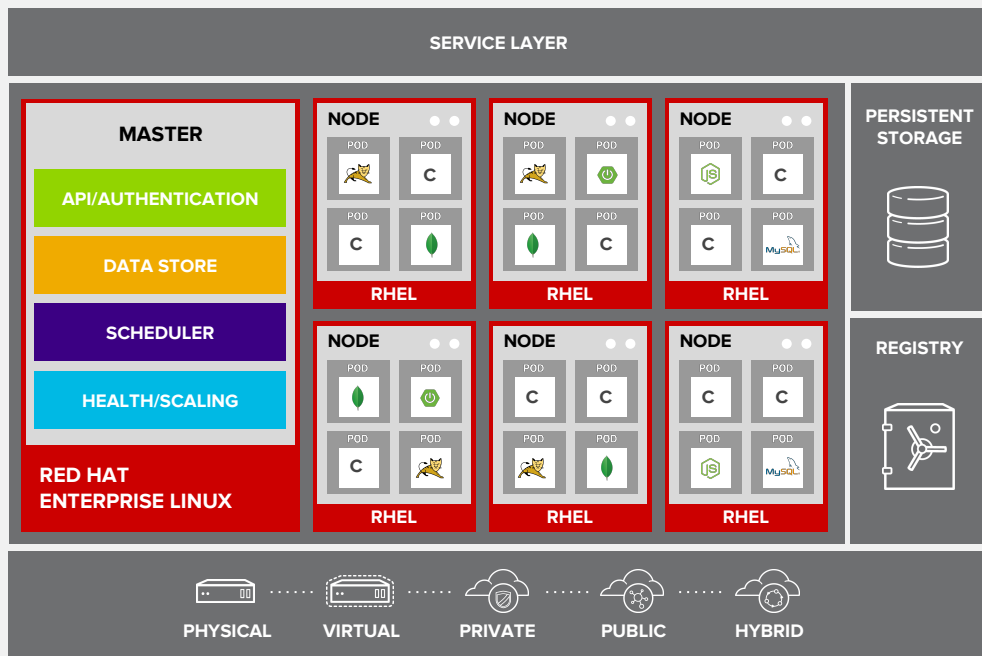
AUTOSCALING PODS



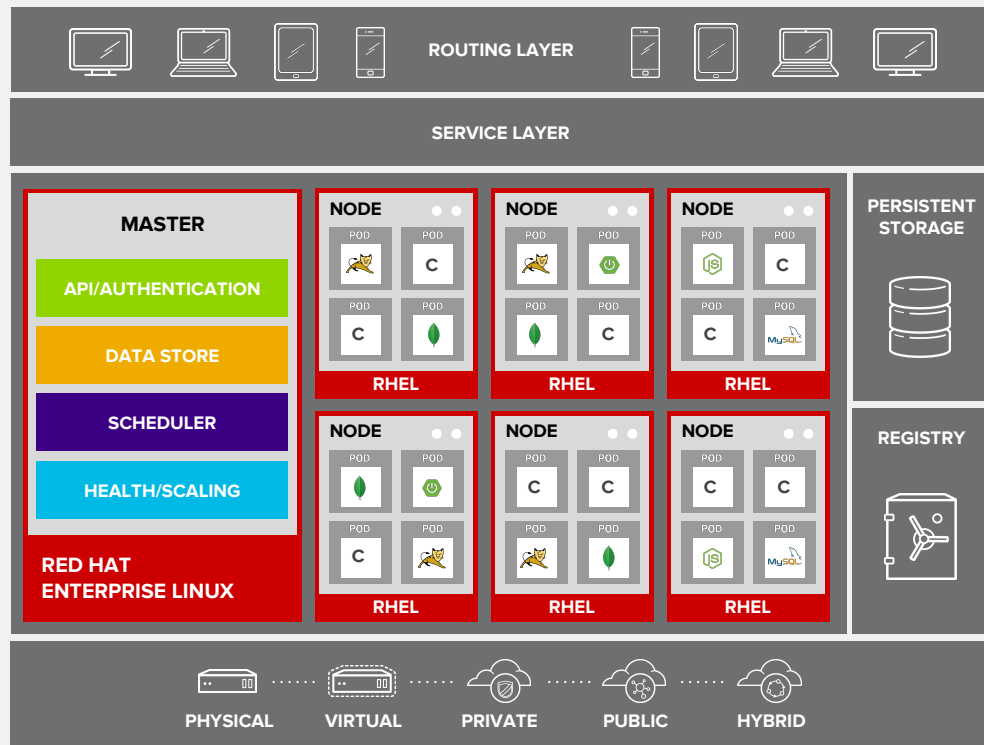
SERVICE DISCOVERY



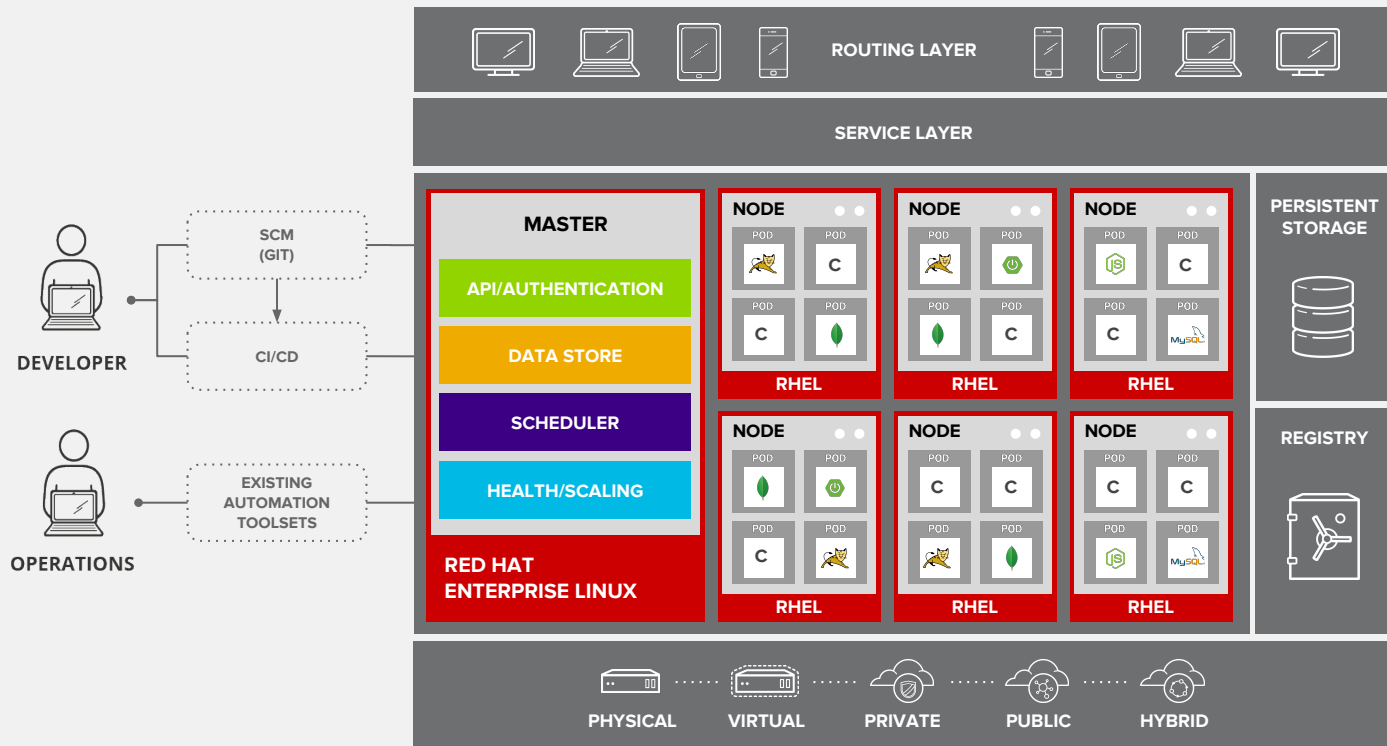
PERSISTENT DATA IN CONTAINERS



ROUTING AND LOAD-BALANCING

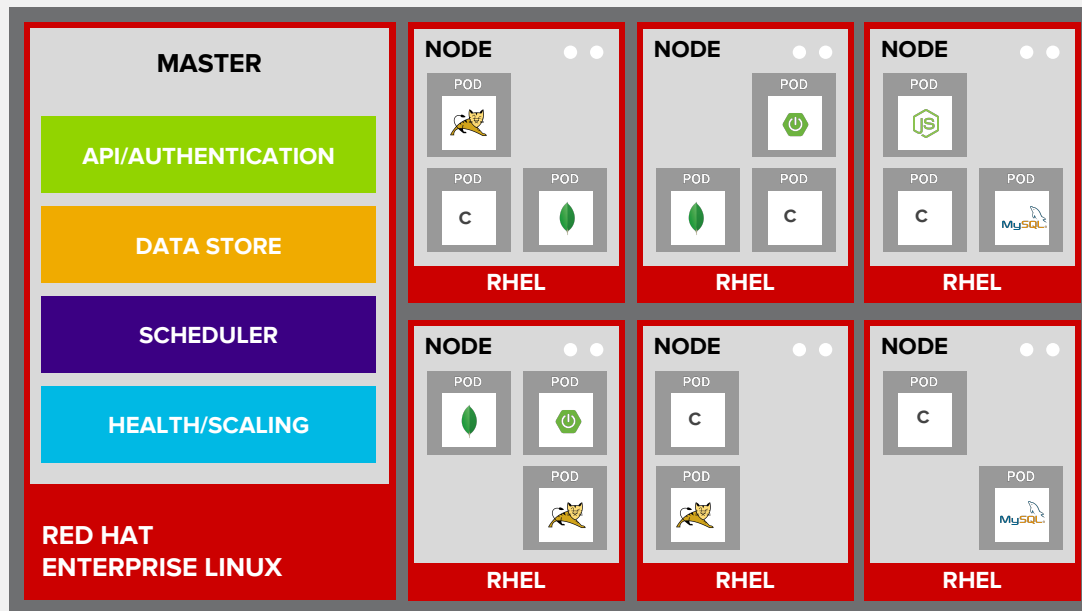


ACCESS VIA WEB, CLI, IDE AND API

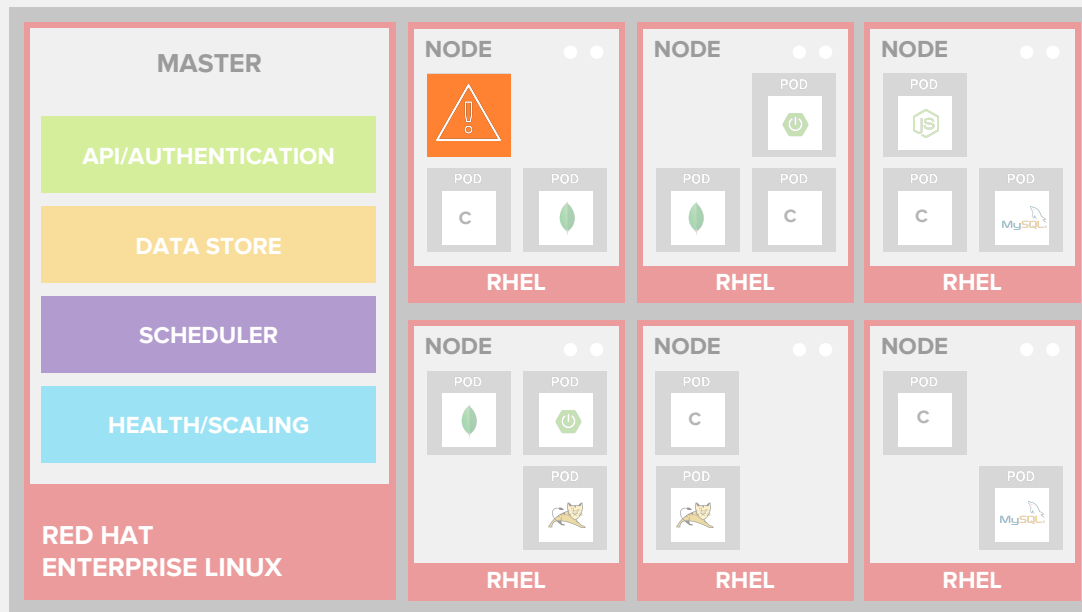


MONITORING APPLICATION HEALTH

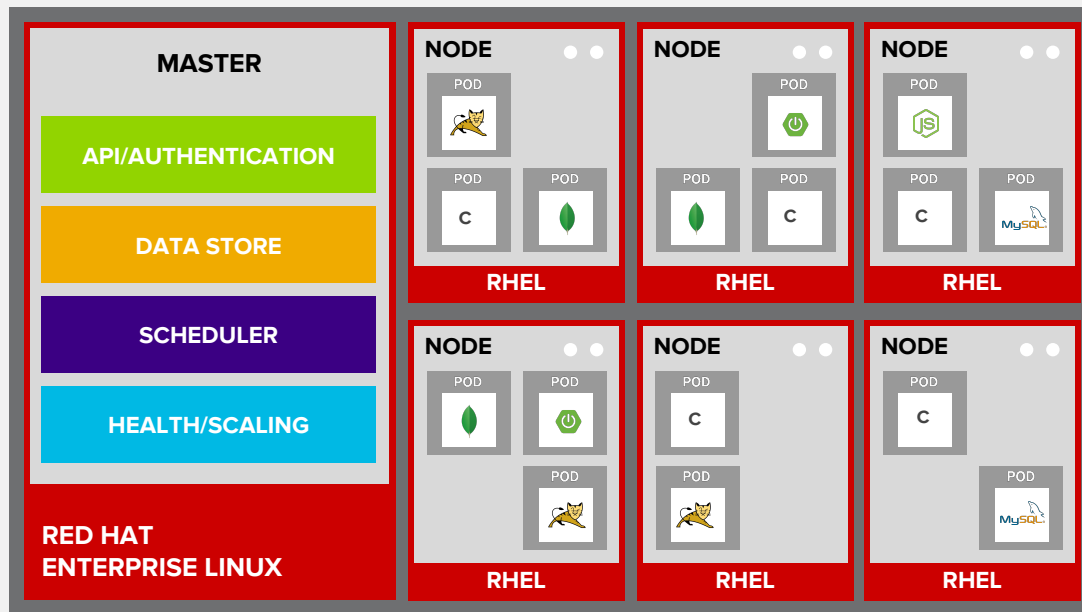
AUTO-HEALING FAILED PODS



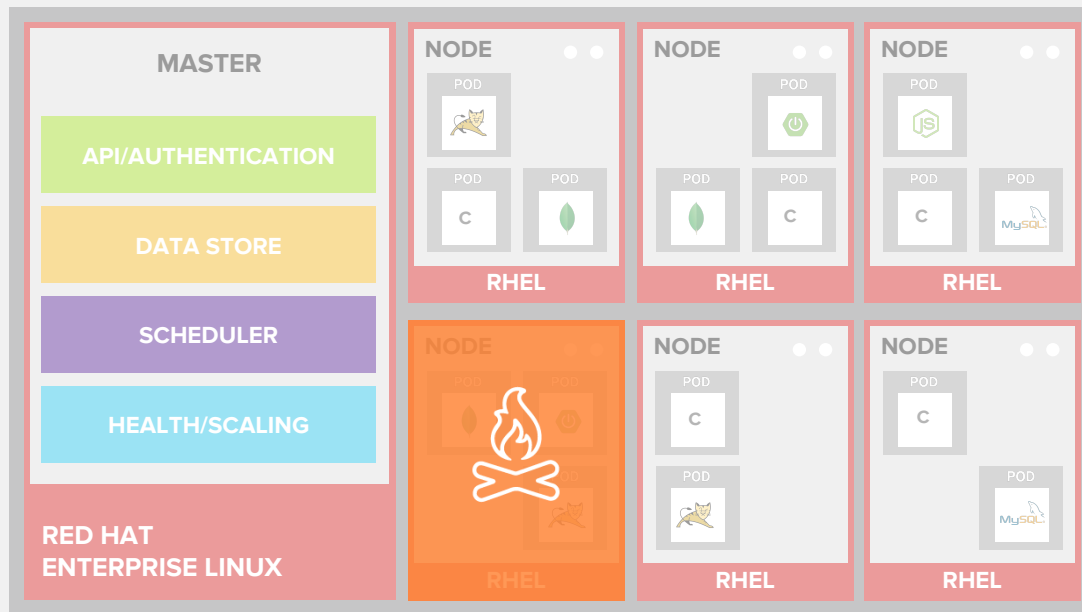
AUTO-HEALING FAILED CONTAINERS



AUTO-HEALING FAILED CONTAINERS



AUTO-HEALING FAILED CONTAINERS



AUTO-HEALING FAILED CONTAINERS

