CMPE 282 Cloud Services
Containers

Instructor: Kong Li

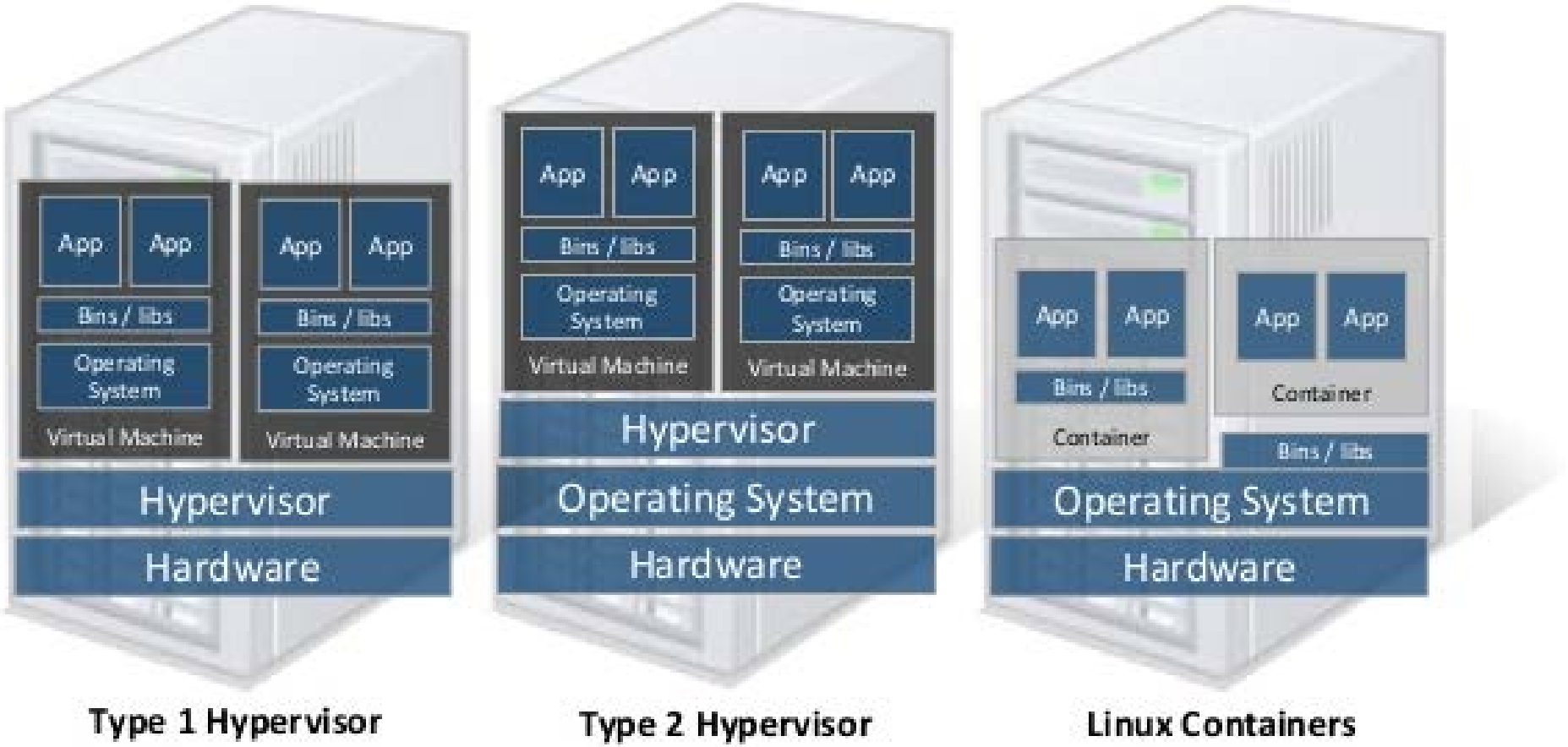
Content

- Containers
- Micro OS
- Docker
- Google Container Engine
- AWS EC2 Container Service
- VMware
- Microsoft
- Others

Containers

- Light-weight application packaging model
  - Isolated runtime env + all dependencies + tools, on top of (& w/o) OS
    - Lighter (and more efficient, less overhead) than VM (why?)
- Simpler deployment/mgmt: No installation/deployment issue
- Resource isolation (cpu, memory, network, process, FS, etc.)
- Running on physical machine or VM; in public/private cloud
  - Usually on top of "micro" OS
- Managed as if they were themselves VMs (scale out/in, migration, etc)
- Application portability: Cure for PaaS vendor's portability issue?
- Platform/runtime for cloud-native apps (microservices)

Containers vs VMs



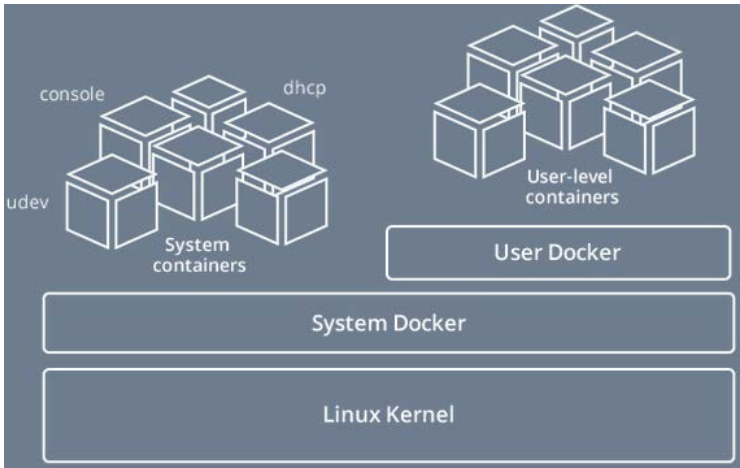
Type 1 Hypervisor **Type 2 Hypervisor** **Linux Containers**

Micro OS

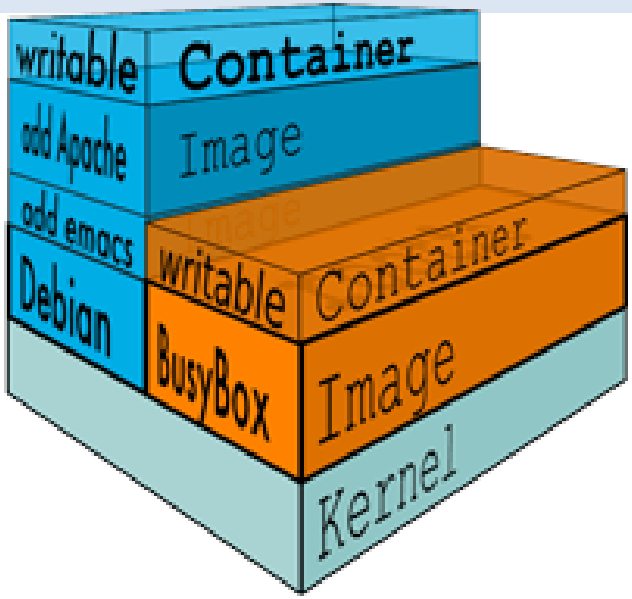
- Designed/optimized for one thing, and one thing only: no GUI
- Smaller footprint/size
  - Less resource consumption
  - less complexity, less patching
  - smaller attack surface, fewer vulnerabilities/reboots, etc
- Cloud/Container OS
- OS update: most are imaged-based (replace the entire OS image), not traditional package-based
- CoreOS Container Linux: coreos.com
  - Linux-based: for docker and rkt container; package service to container
  - Image-based update: ~161 MB
- Red Hat Project Atomic: www.projectatomic.io
  - Atomic host: Linux-based - Fedora, RHEL, and CentOS
  - Image-based update: a few hundreds MB

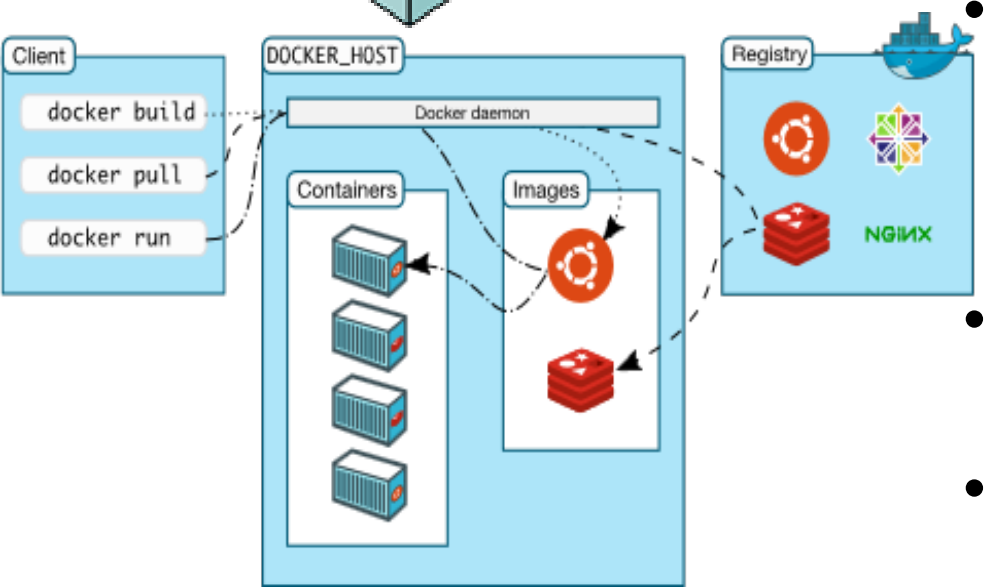
Micro OS (cont'd)

- Ubuntu Core: www.ubuntu.com/core
  - Image-based update: ~100MB
- VMware Photon OS: vmware.github.io/photon
  - Linux based (~260MB), integrated with vSphere, VMware ecosystem
  - Container: Docker, rkt, Garden (Pivotal's Cloud Foundry)
  - Update: image-based and traditional package based
- Rancher Labs RancherOS: rancher.com/rancher-os
  - System docker (PID 1): start system processes *as containers*
  - User docker: user-level containers
  - Run everything in container; ISO: ~55MB
- Microsoft Nano Server
  - No GUI, CLI, etc. All mgmt done remotely
  - Docker



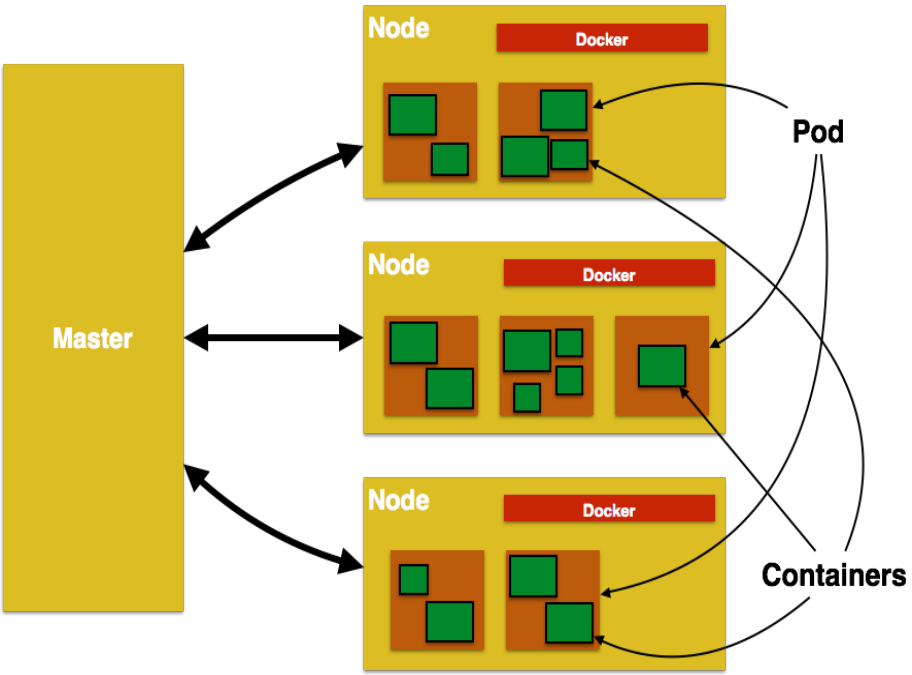
Docker



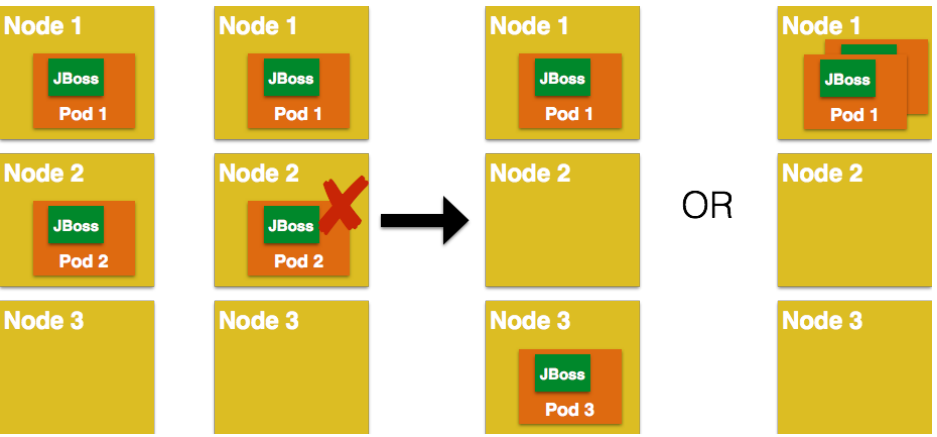


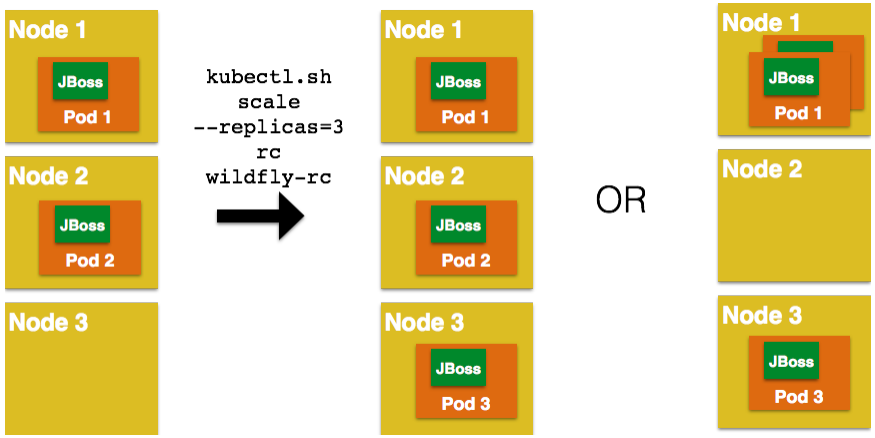
- Docker Engine
  - Based on Linux container LXC
  - Images (left)
    - Union File system: Multiple layers of FS + writable layer
  - Share/deploy faster, scale easily
  - Docker Hub, Docker Registry (left)
- Built-in orchestration – Swarm
  - cluster, scheduling, placement
  - Integrated or standalone: since v1.12
- Docker Machine: auto provision hosts across multi-platforms
- Docker Compose: define multi-container apps

Google Container Engine (GKE)

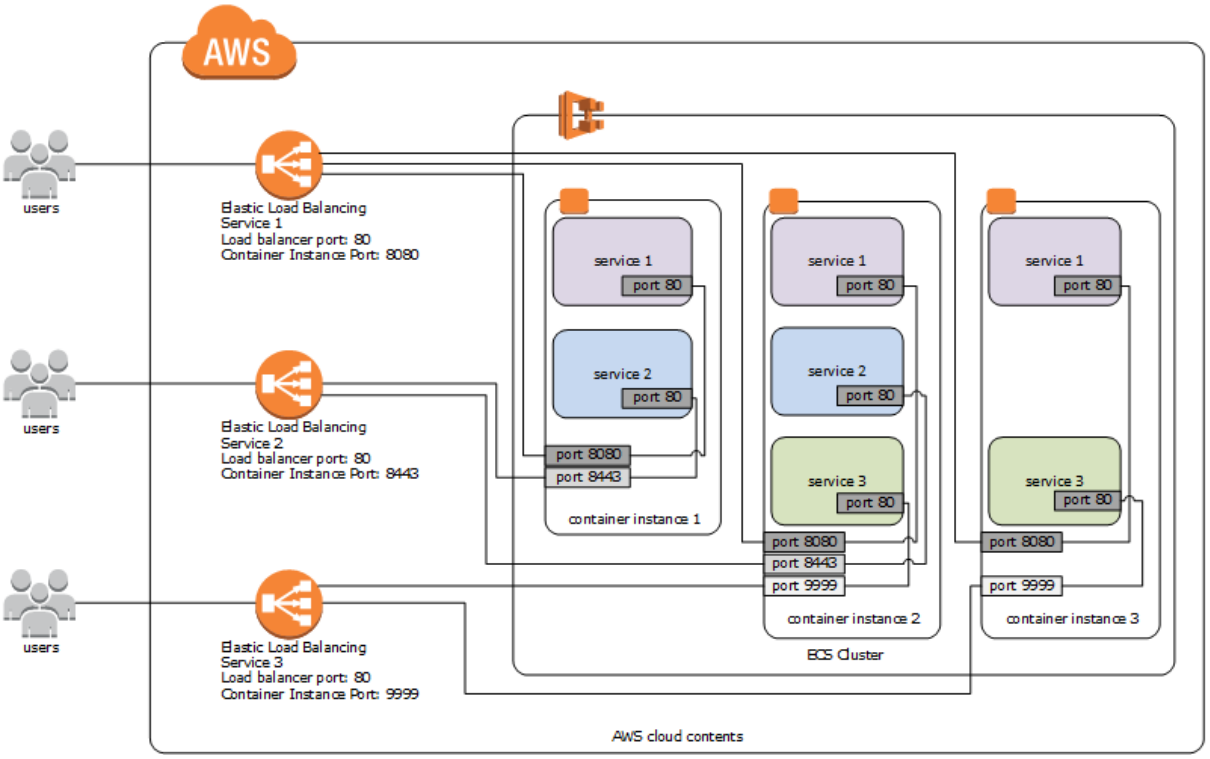


- Docker support
- Run on a cluster of GCE (VMs)
- Kubernetes: managing containerized apps across multi-hosts
  - Logically organize containers to groups (left)
  - Replication: optimal # of pods
    - Rescheduling pods (lower left)
    - Scaling pods (lower right)
  - Cluster Auto-restarting
  - Docker & rkt support
  - GCE, AWS, Azure, vSphere, etc.





AWS EC2 Container Service (ECS)



- Docker support

Across a cluster of EC2 instances

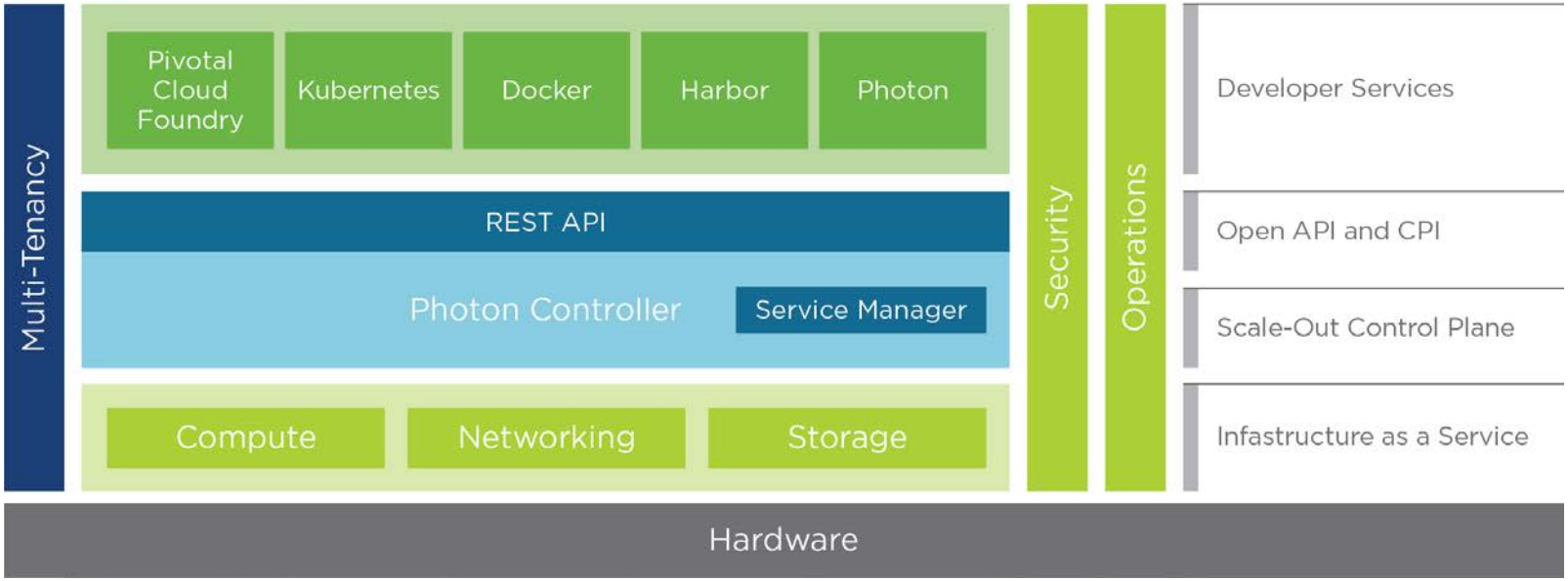
  - App span multiple AZs
  - Placement mgmt: auto or manual control
  - Cluster mgmt and config mgmt
  - Scheduling: balance between resource needs and availability requirements
  - Integrated w/ ELB (left)

- Docker workload can be migrated to/from AWS

VMware Photon Platform

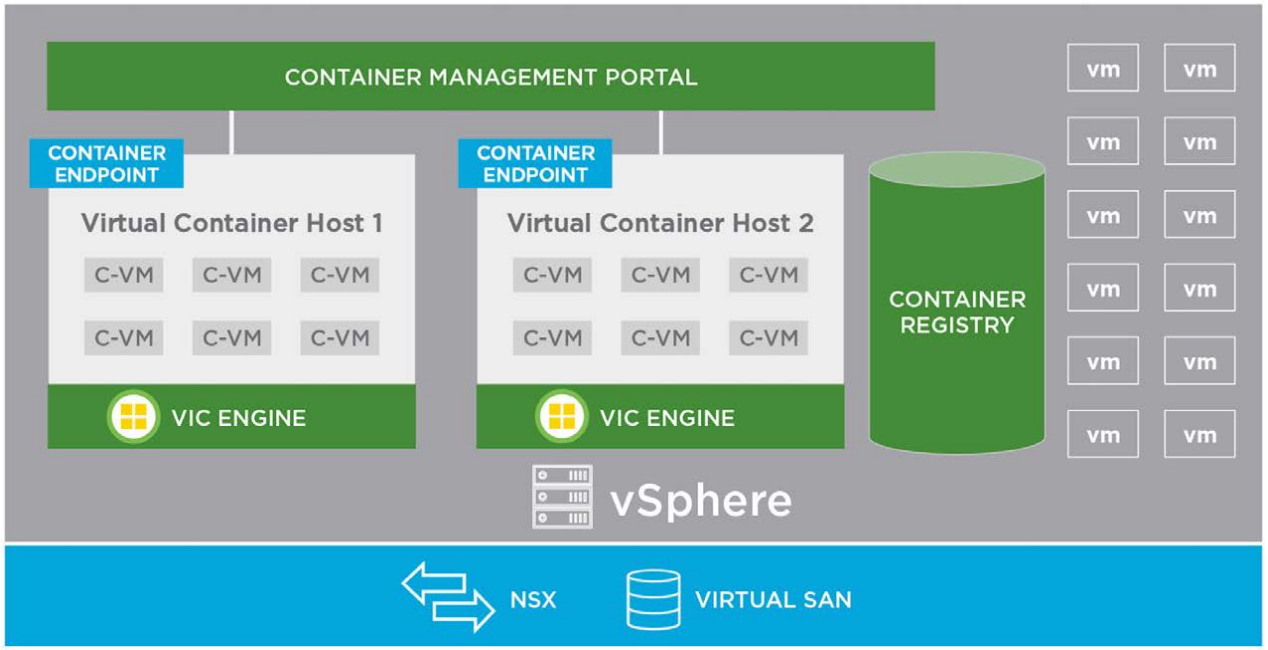
- Photon Platform: container hosting env
  - Photon Machine: Secure ESX Microvisor + Photon OS
  - Photon Controller: Distributed/clustered/multi-tenant mgmt plane
  - Full integration with Kubernetes, Docker, Pivotal Cloud Foundry
  - No vCenter server is needed; no vMotion/DRS
  - Focus on scale and speed

VMware Photon Platform



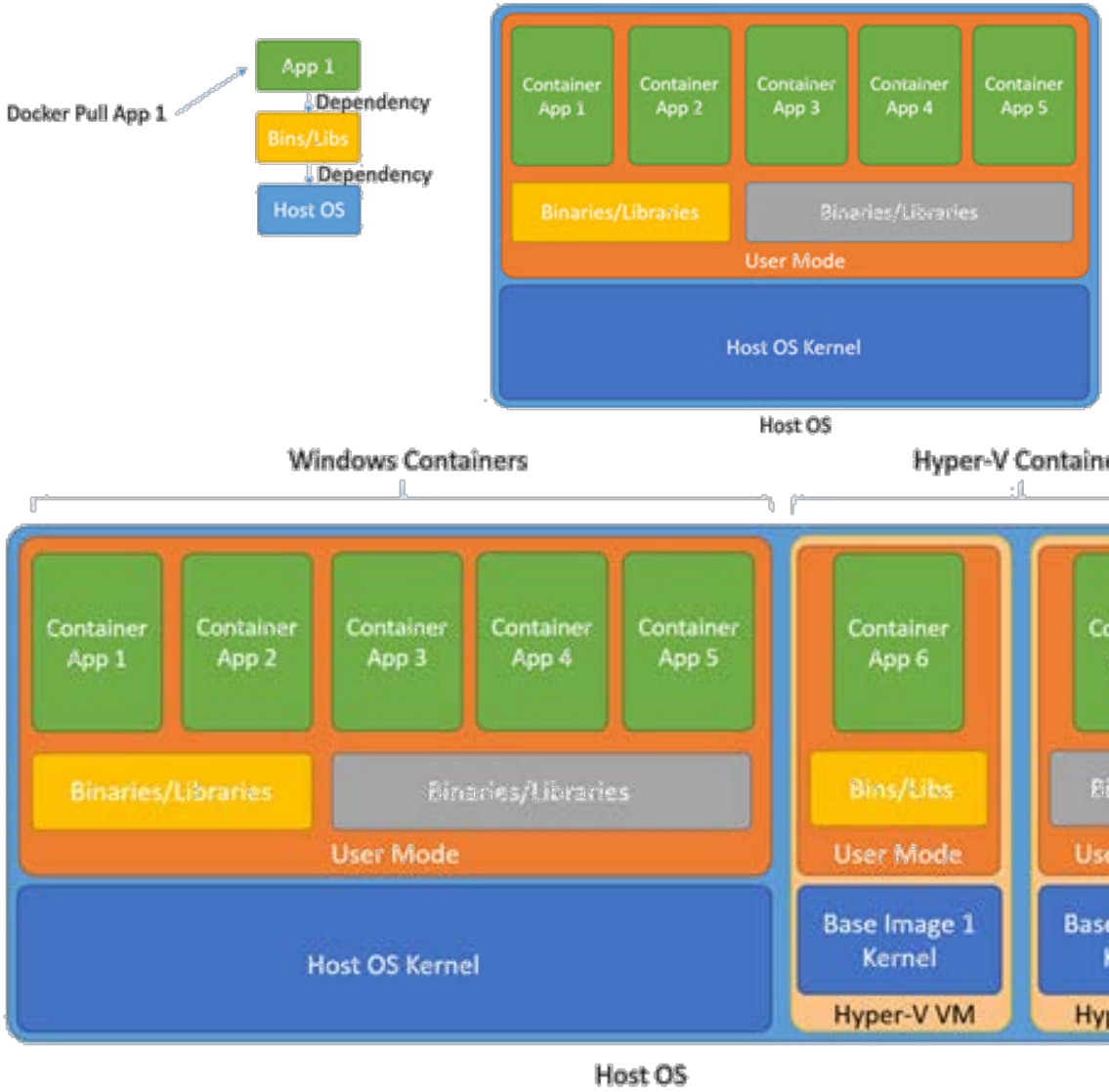
VMware vSphere Integrated Containers (VIC)

- VIC: run “container as VM” instead of “container in VM”
  - Extend VM mgmt to container mgmt
- Virtual Container Host (VCH)
  - container endpoint w/ dynamic boundaries; Single-tenant
  - A VCH is an vApp running multiple containerVM (C-VM)
  - A ESXi host may have multiple VCHs, along with other regular VMs



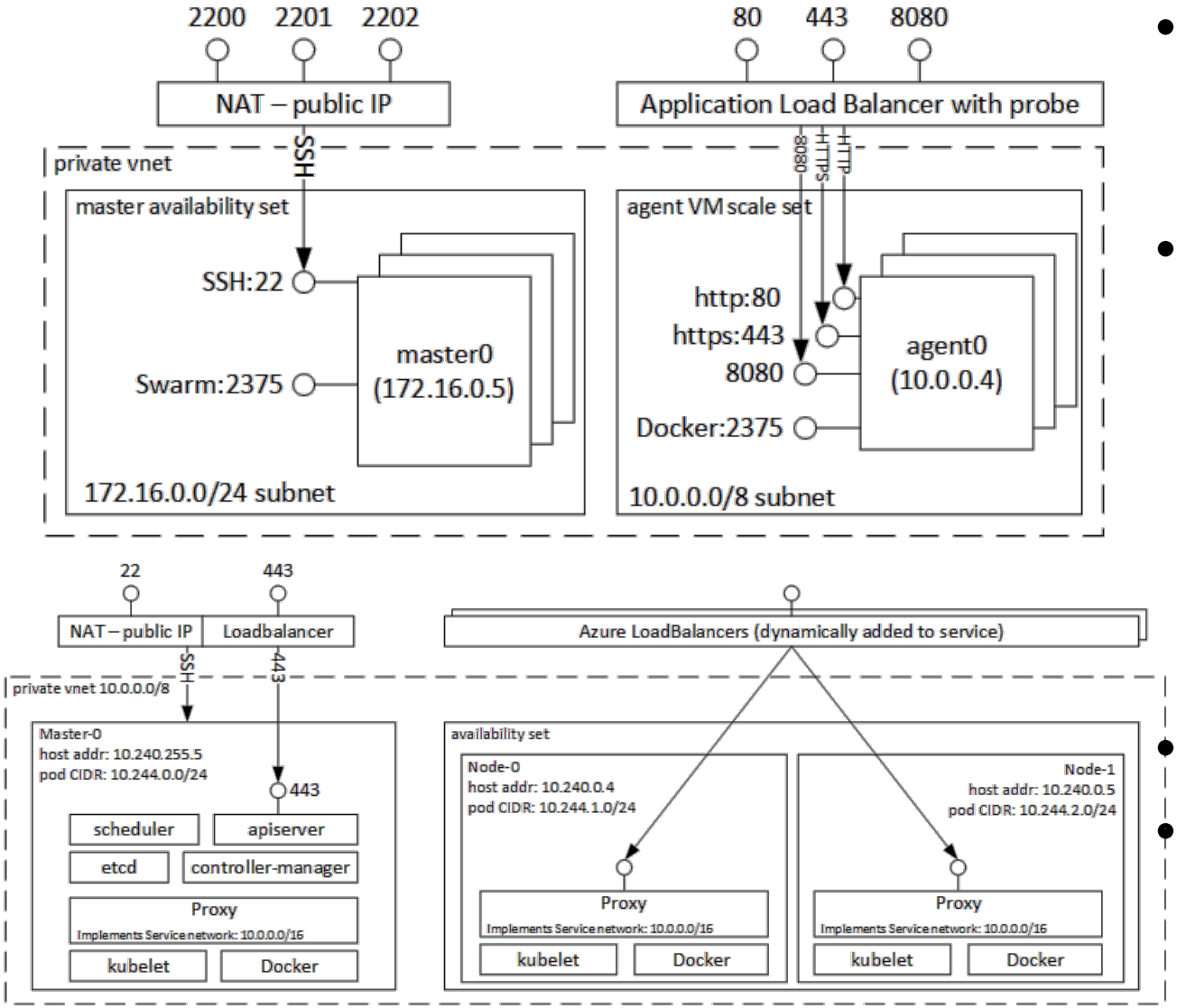
- VIC
  - Multi-tenancy
  - Dynamic resource allocation & container placement vSphere resource pool
  - Focus on security & mgmt

Microsoft Windows/Hyper-V Container



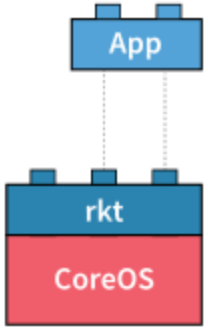
- Nano server
- Windows 2016 server:
  - Windows container
  - Hyper-V container
    - Deployment time: pick isolation level - Windows or Hyper-V container
    - Docker and Apache Mesos

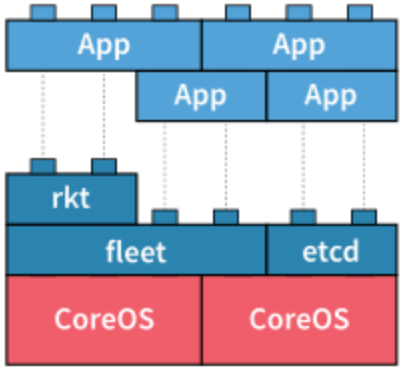
Microsoft Azure Container Service (ACS)



- Container hosting env for Docker: Linux, windows
- orchestrator + cluster tools: Docker Swarm (upper left), Kubernetes (lower left), DC/OS (Apache Mesos), Azure Service Fabric
- API: REST
- Azure Container Instance (ACI): no K8s required

Others

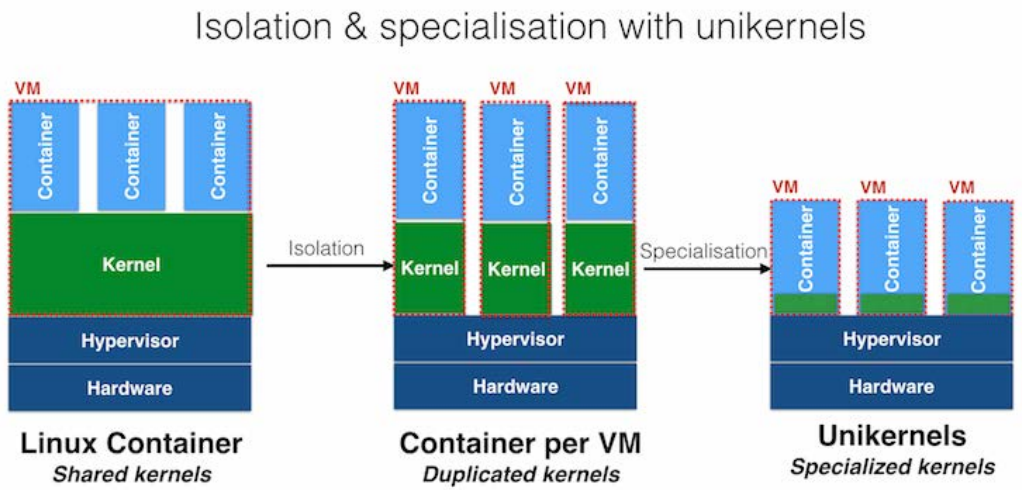




- CoreOS – composable tools
  - CoreOS: light-weight OS
  - Rkt: container runtime
    - can run Docker images
    - Implementation of App Container Spec
  - Fleet: cluster mgmt
  - Etcd: distributed, key-value store
- Red Hat
- IBM
- Open Containers Initiative
  - https://www.opencontainers.org/
- App Container Spec
  - https://github.com/appc/spec

Unikernels

- Def: customized, single address space OS == {app source code} + {fixed-purposed kernel that includes only the functionality required by the app}
  - Run on hardware or hypervisor
  - vs Micro OS: Tighter integration between app and OS
- Pros:
  - Improved security
  - smaller footprint
  - Better efficiency
  - faster boot time
  - whole-system optimization
- Cons:
  - Not for general purpose
  - Not for multi-user
  - Difficult to add functionality
- http://unikernel.org/projects/
- Docker acquired Unikernel Systems



References

- Docker: https://www.docker.com
- Kubernetes
  - http://kubernetes.io/
  - http://blog.arungupta.me/kubernetes-design-patterns/
- AWS EC2 Container Service: https://aws.amazon.com/ecs
- VMware
  - Photon Platform: https://www.vmware.com/products/photon-platform.html
  - vSphere Integrated Container: https://www.vmware.com/products/vsphere/integrated-containers.html
  - http://blogs.vmware.com/vsphere/2015/08/how-to-choose-the-best-infrastructure-stack-for-your-cloud-native-applications.html
- Google Container Engine: https://cloud.google.com/container-engine
- Microsoft Azure container service: https://docs.microsoft.com/en-us/azure/container-service/

References (cont'd)

- CoreOS: https://coreos.com/
  - Rkt: https://github.com/coreos/rkt
- http://www.ruurdkeizer.com/
- Open Containers Initiative: https://www.opencontainers.org/
- App Container Spec: https://github.com/appc/spec
- Unikernels
  - https://en.wikipedia.org/wiki/Unikernel
  - https://blog.docker.com/2016/01/unikernel/