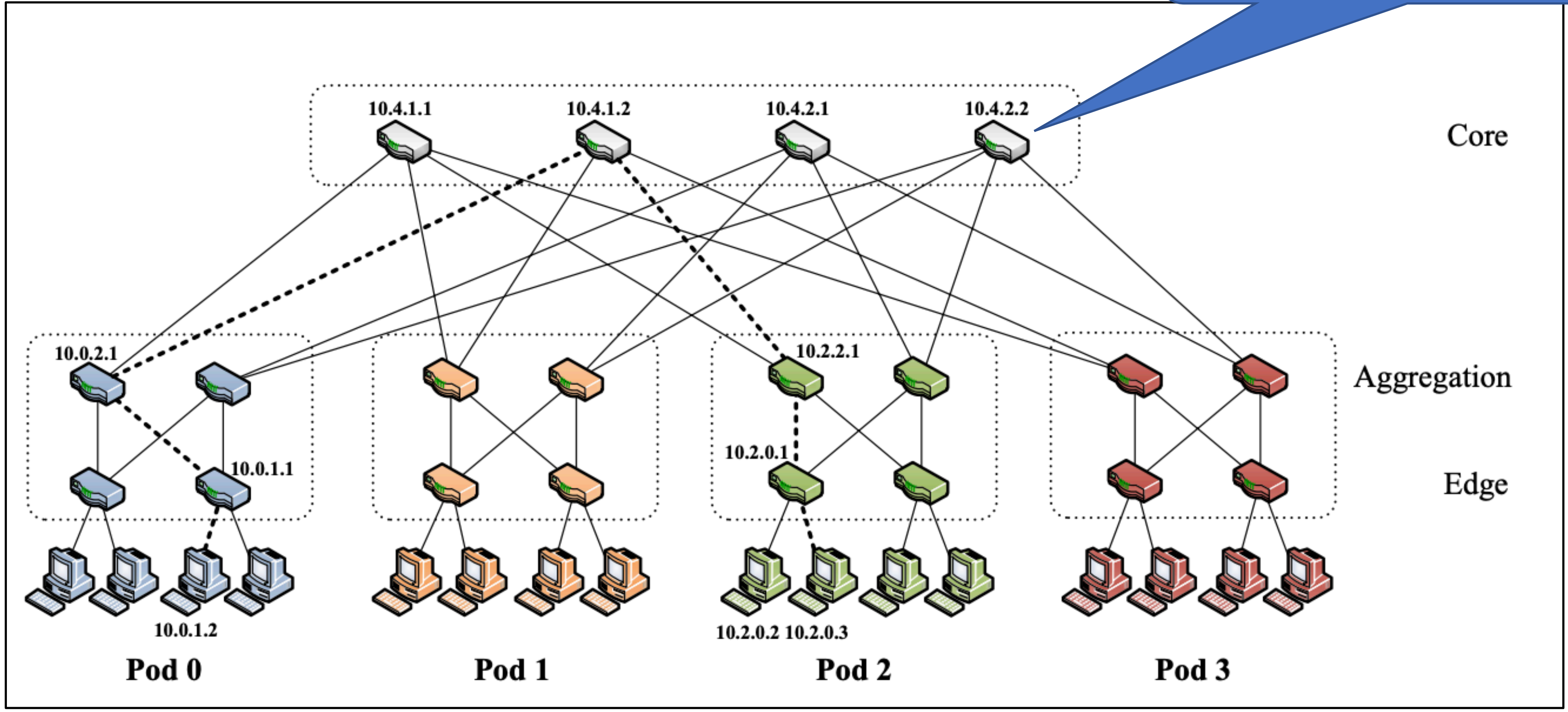
Networking on the ground, in the cloud, and in containers



What is in the box?

Router

A computer optimized for routing and forwarding

- Operating system to manage resources
- Routing protocol implementations (e.g., BGP, OSPF)
- Lots of ports (not TCP ports)
- Chip to forward traffic between ports at “line rate”

Router (2)

Traditionally, a hardware-software combo sold by a router vendor

- Cisco
- Juniper
- Arista
- ….

But moving toward open systems

- SONiC – open source router OS from Microsoft
- Running on commodity hardware

Configuring the router

Routers are not plug-n-play

- Configure IP addresses
- Configure which protocols to run
- Configure those protocols
- Configure management aspects, e.g., DNS servers, NTP servers

Configuration uses custom syntax:

- Example Cisco file: https://github.com/batfish/pybatfish/blob/master/jupyter_notebooks/networks/example/configs/as1border2.cfg

Configuring the router (2)

Traditionally, configuration has been done manually

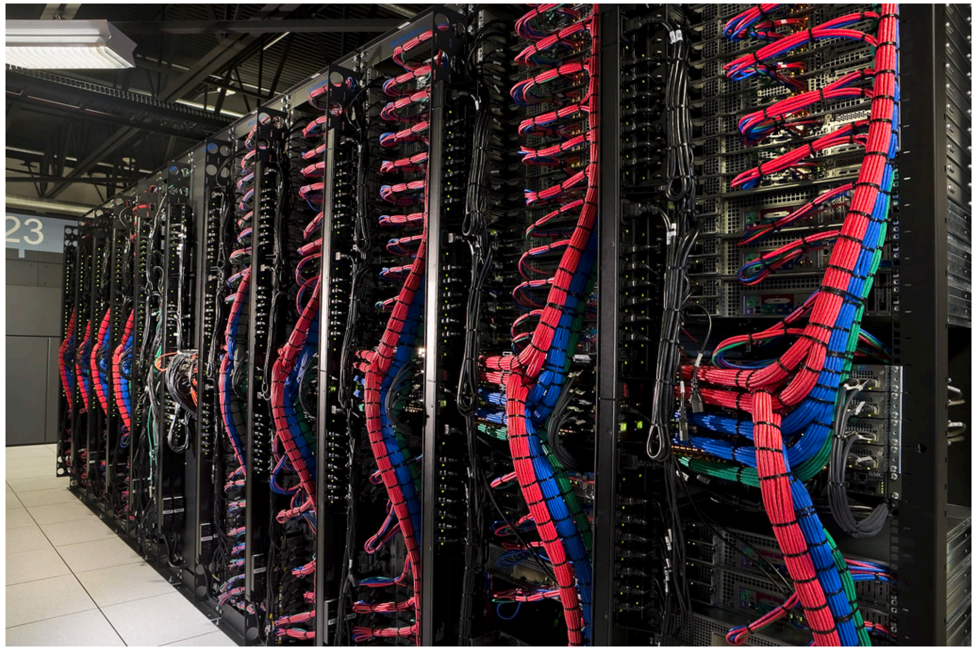
- Figure out the change, reason about it manually
- Log in to the router and apply the change
- High risk of logical errors and “fat fingers”

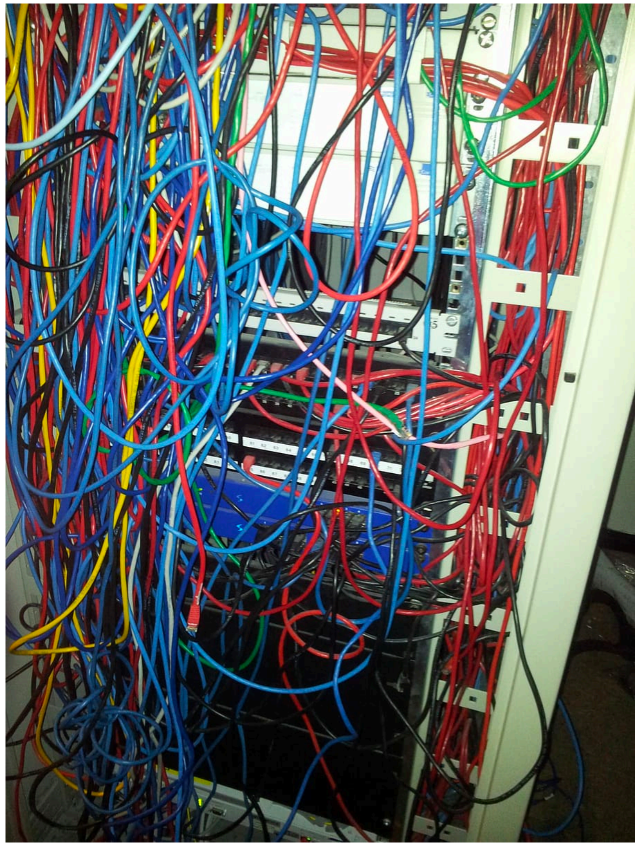
Increasingly, more automation

- Ansible, SaltStack, Nornir
- Batfish

Making a network out of routers

1. Get them connected





Making a network out of routers

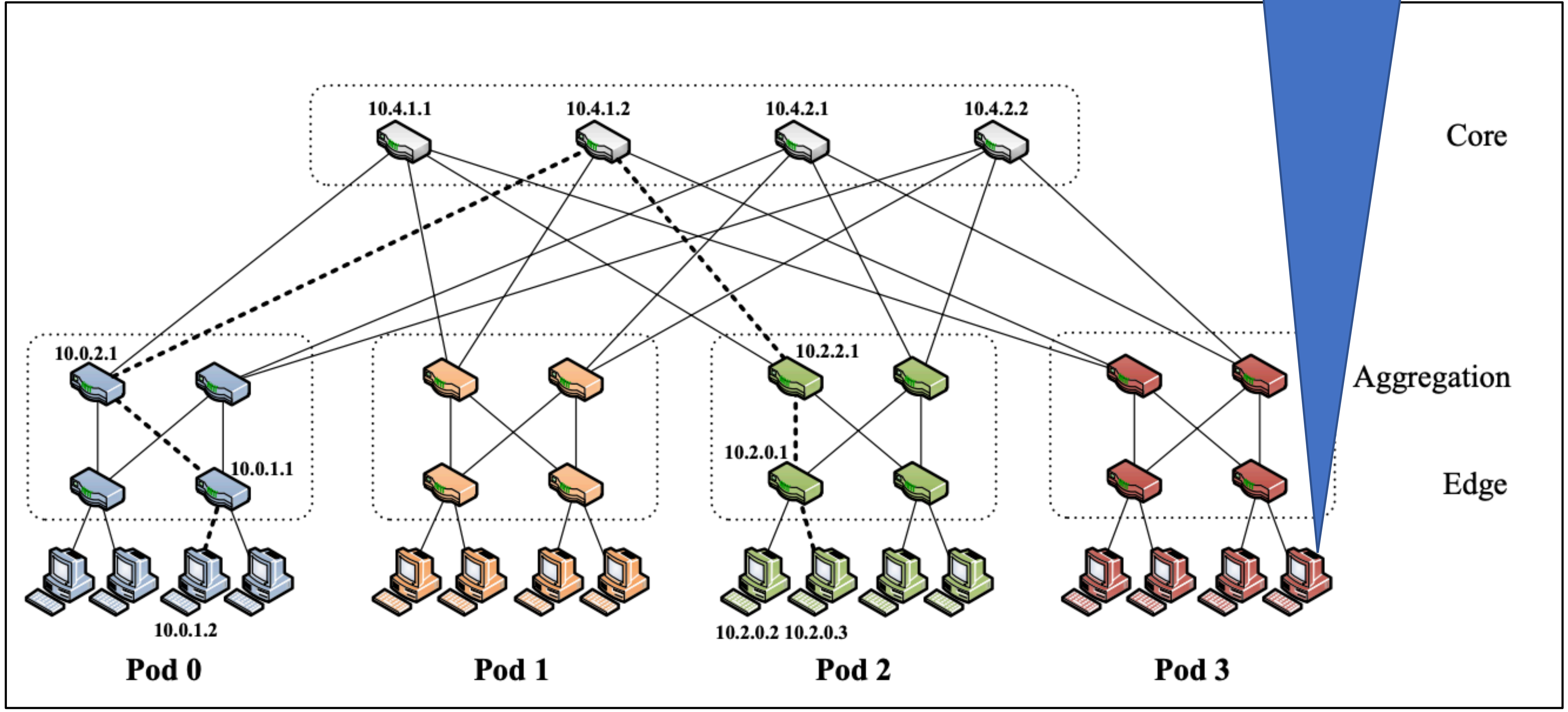
1. Get them connected

2. Configure routers
    - Basic initial configuration provides connectivity to the router

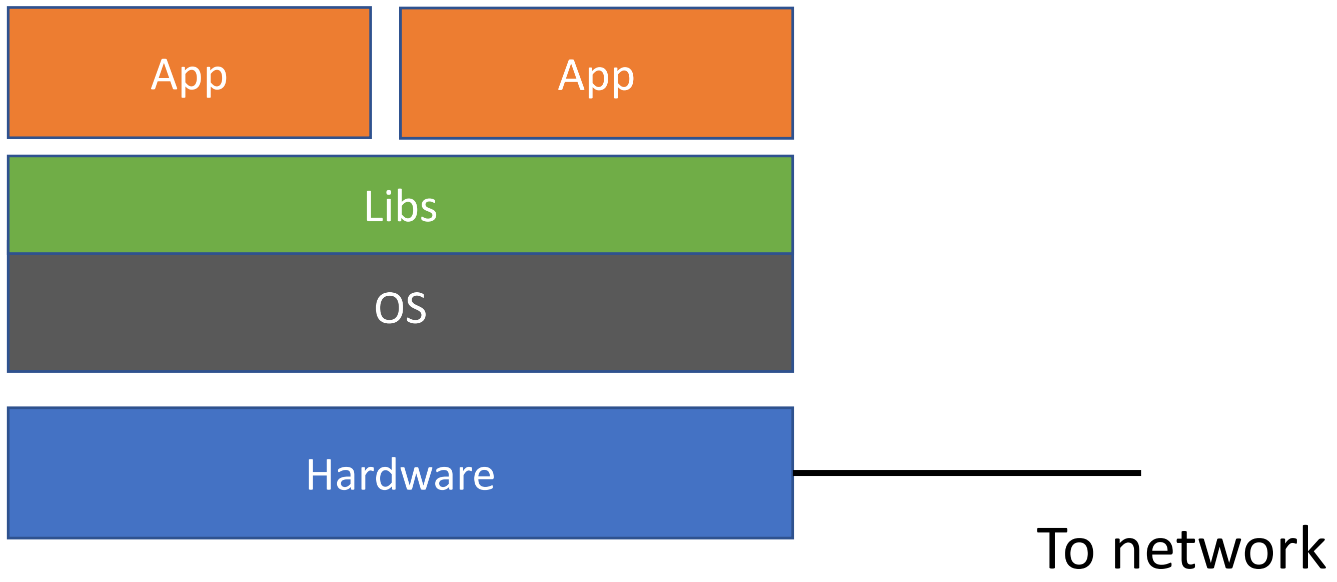
3. Monitor, monitor, monitor

4. Configuration changes and maintenance

What is in this box?



Originally



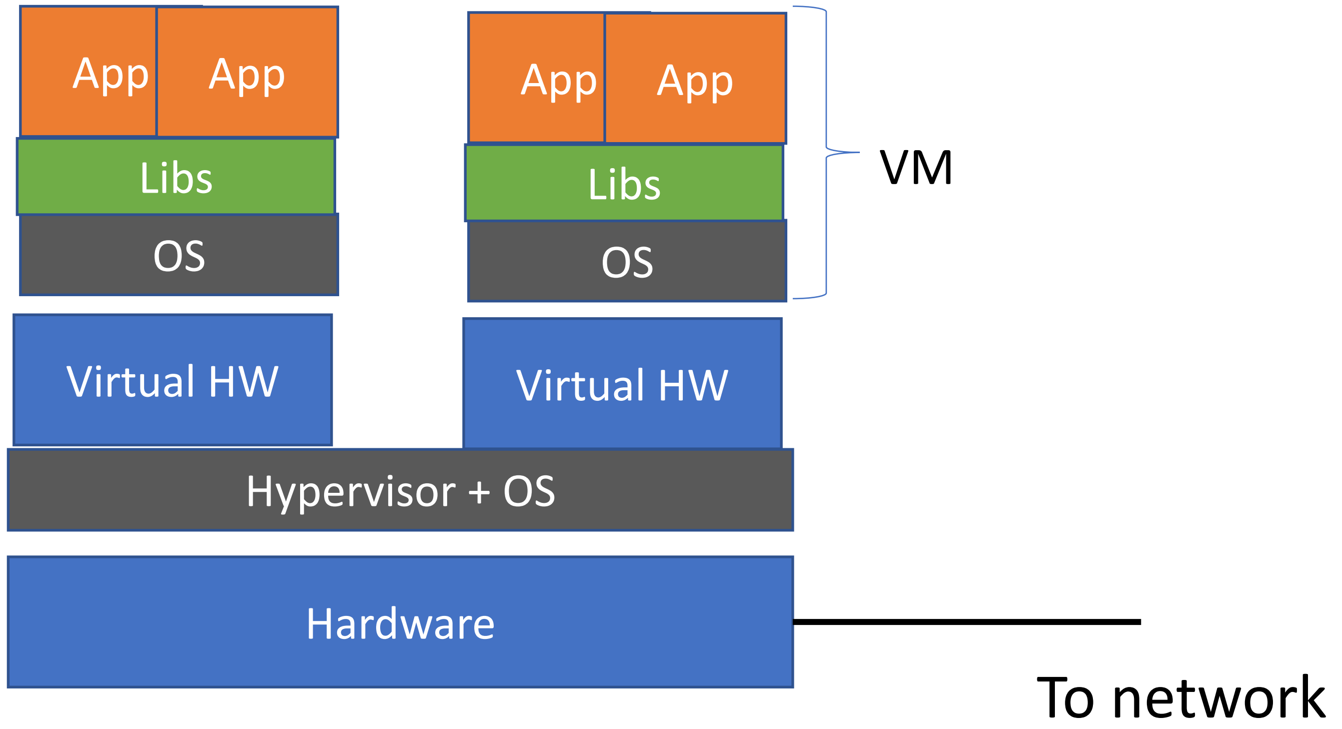
Then came virtual machines (VMs)

HW became too powerful

- Run multiple OSes on the same machine
- Cheaper that way

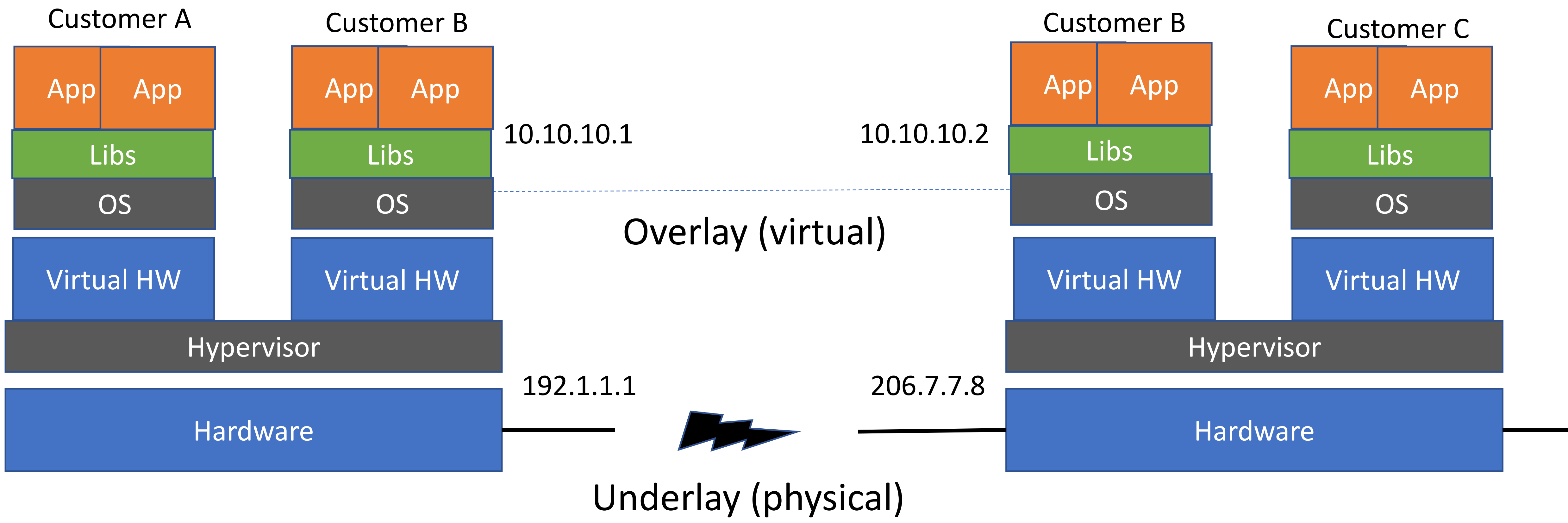
The hypervisor virtualizes the HW and fools the OS

- Provides isolation



The network thinks multiple hosts are connected
The hypervisor acts as a hub for inter-VM traffic

VMs in the cloud



Forwarding between VMs involves a DNS-style lookup from overlay address to underlay location

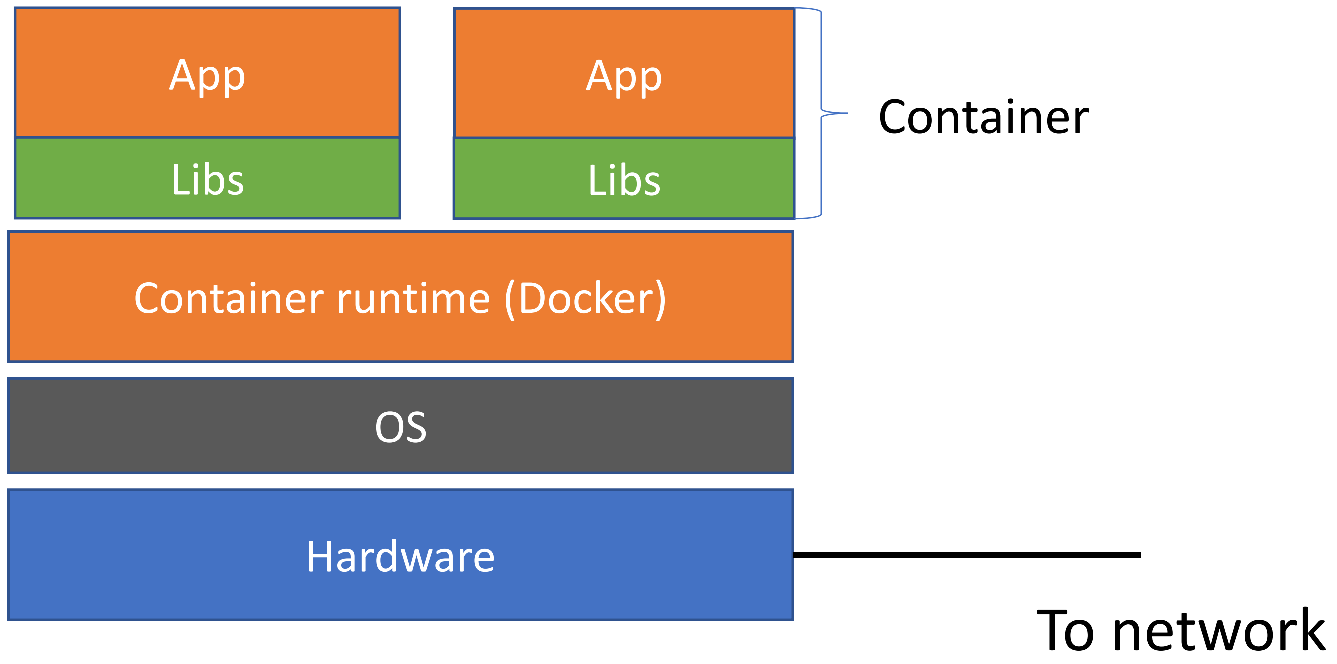
Enter containers

Lighter-weight virtualization than VMs

- Libraries, not the full OS

Better isolation and packaging than apps

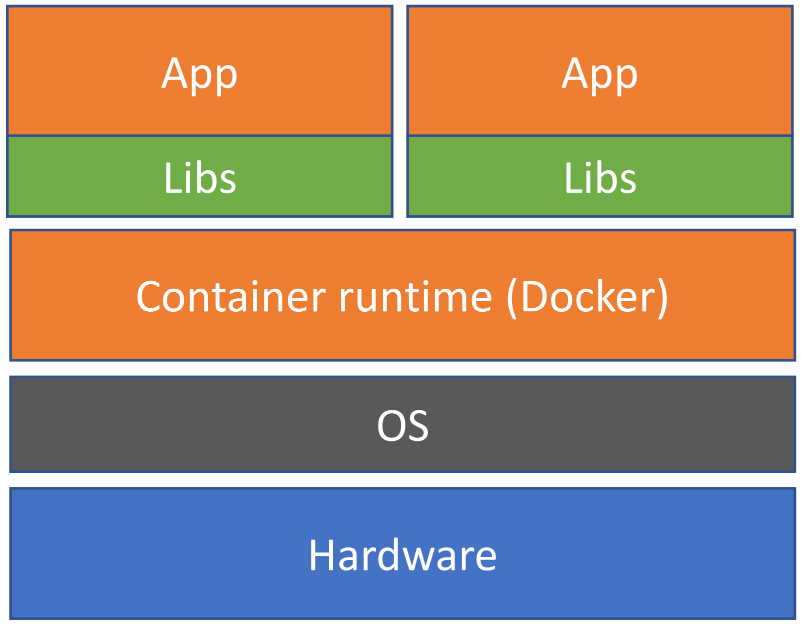
- Bundle the library versions you need



Container networking

Connect containers to the outside world and to each other

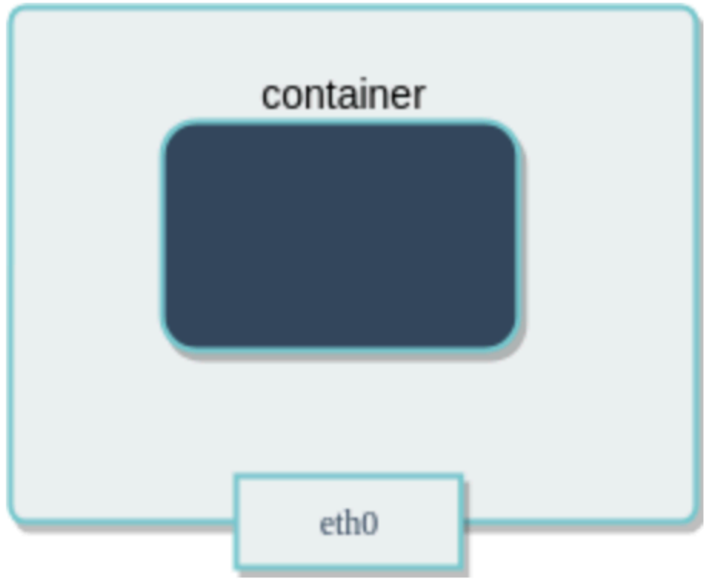
- Port conflicts among containers and other apps running on the same host
- High performance between containers on the same host
- (Virtual) private network between related containers (service mesh)



Container networking: Host

Containers share the IP address (and networking stack) of the host.

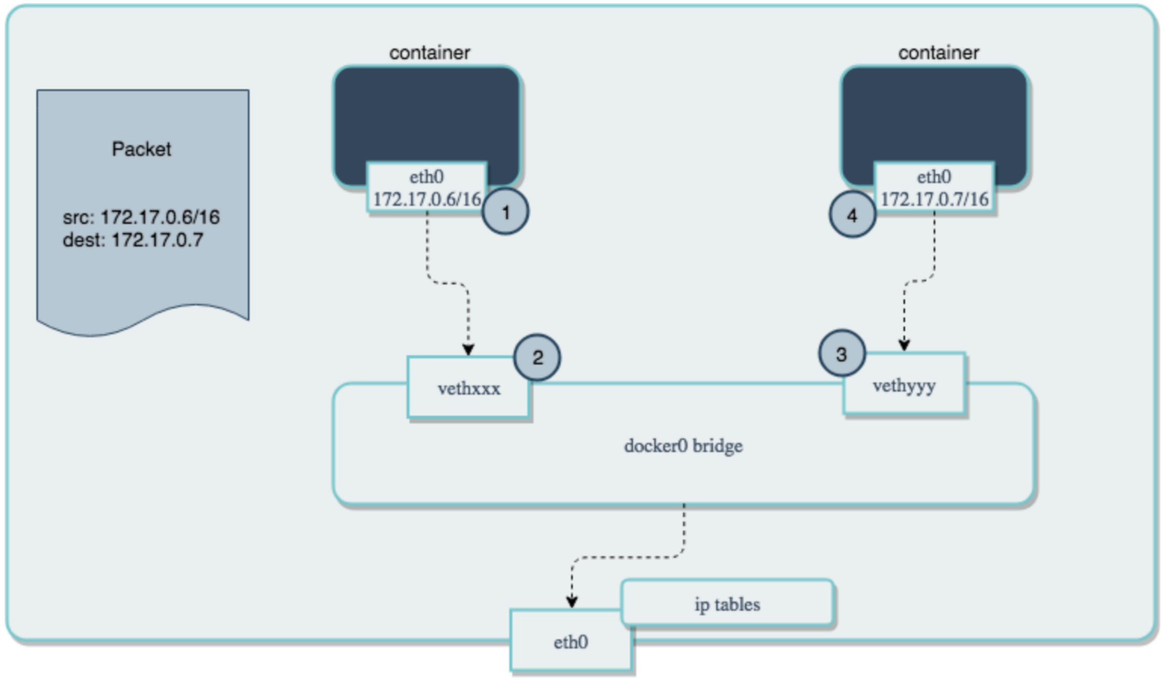
- Cannot handle port conflicts
- Minimal overhead



Container networking: Bridge

An internal network for containers on the same host.

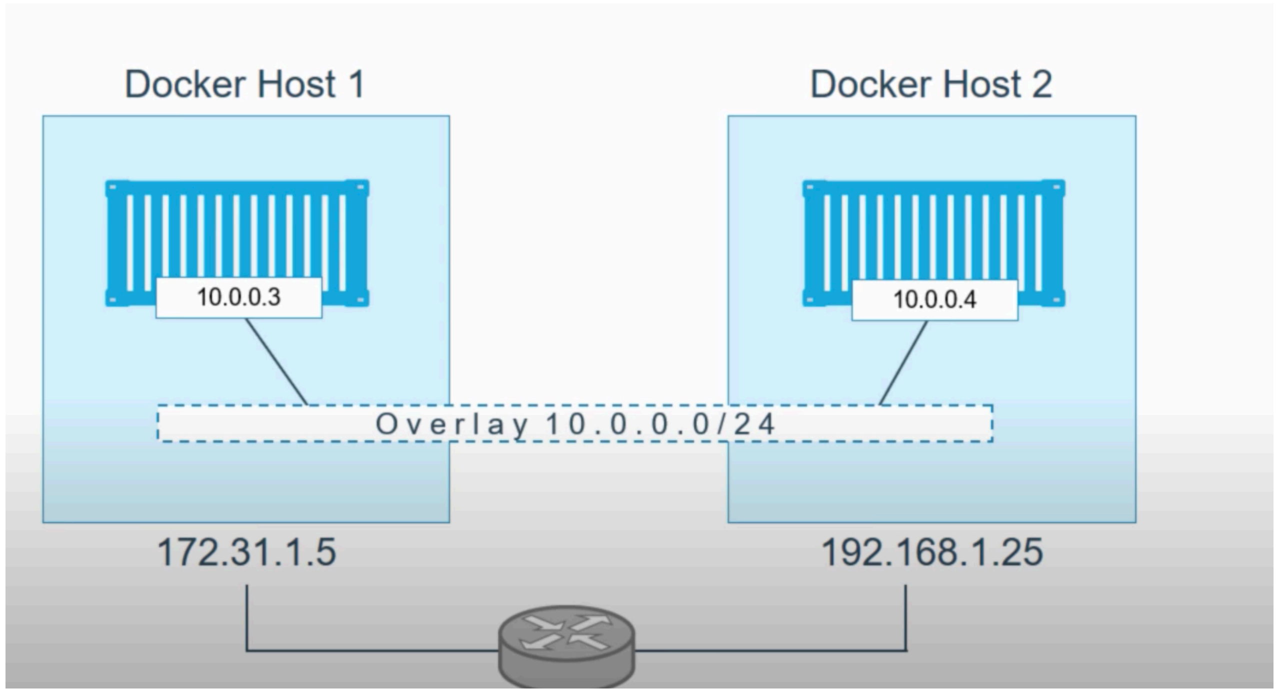
- Use NATs for outside world



Container networking: Overlay

Create a private network across containers on different hosts

- VXLAN is a common way to do that



CNI: Container networking interface

Specification for writing plugins to configure network interfaces

- Decouple runtime from network configuration
- Plugins provide an interface that orchestration engines can use
- GitHub repo: https://github.com/containernetworking/cni