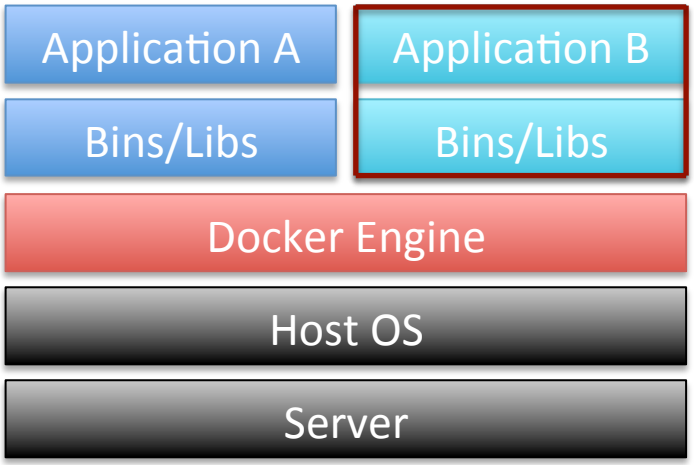
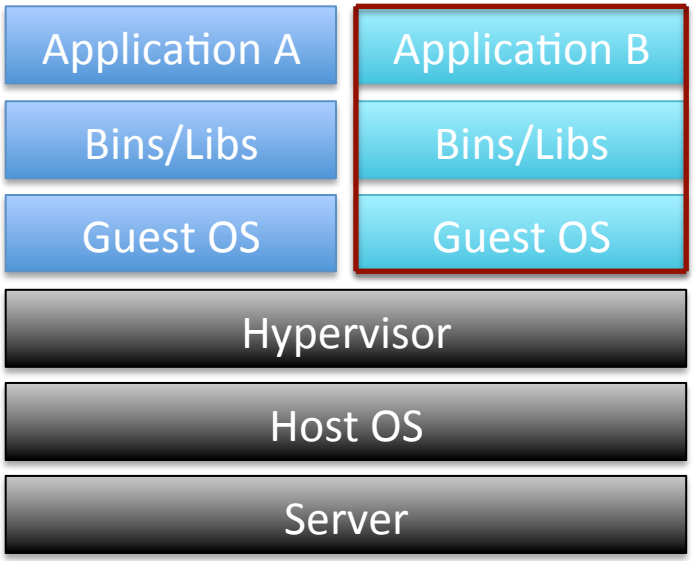
Containers and IPv6

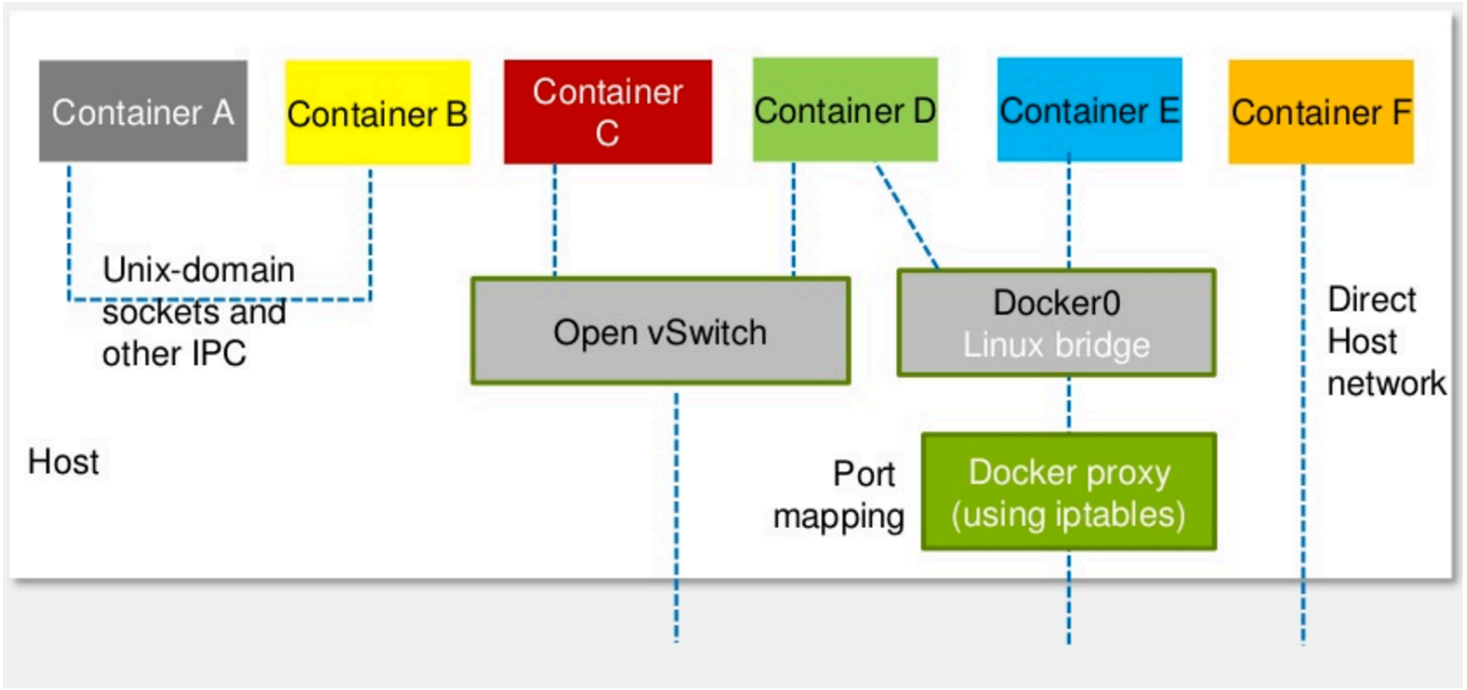
Stephen Youell

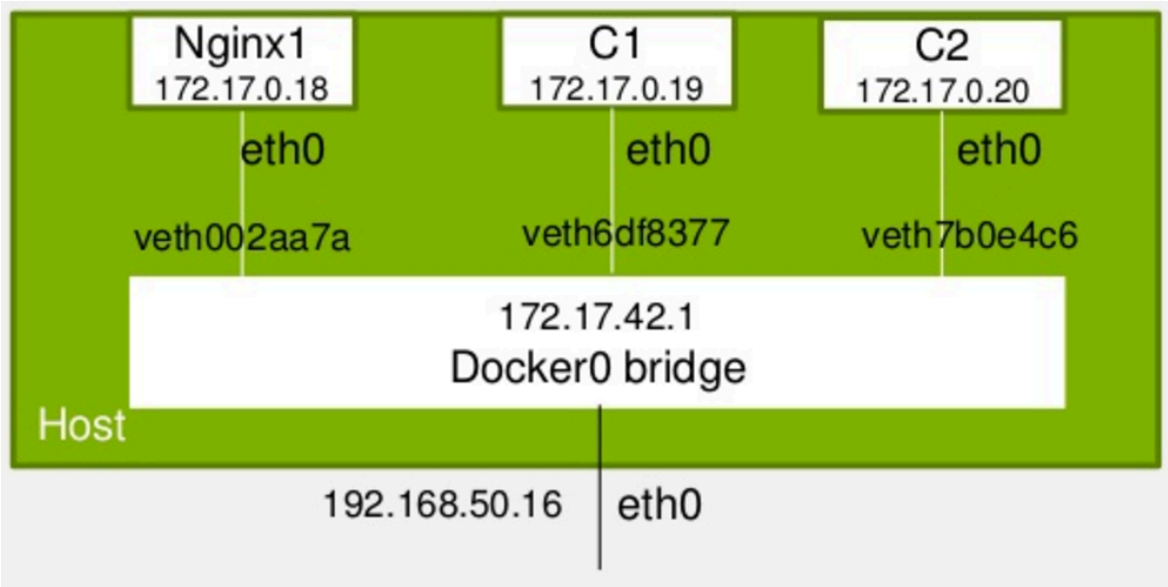
Docker Container



Virtual Machine




Container A
Container B
Container C
Container D
Container E
Container F
Unix-domain sockets and other IPC
Open vSwitch
Docker0
Linux bridge
Direct Host network
Host
Port mapping
Docker proxy (using iptables)


Nginx1
172.17.0.18
C1
172.17.0.19
C2
172.17.0.20
eth0
eth0
eth0
veth002aa7a
veth6df8377
veth7b0e4c6
172.17.42.1
Docker0 bridge
Host
192.168.50.16
eth0

- From Docker 1.5 IPv6 is supported
- IPv6 desirable to avoid port forwarding
- One deployment scenario sees containers used for short lived jobs and destroyed – very high number of containers over time

- Configuration highly static
- Multiple ways to configure IP prefixes:
  - Assign new /64 prefix to Docker
  - Assign longer prefix from /64 assigned to host
  - Use addresses from the host subnet with NDP proxy

- Limited operational experience to understand which are best deployment scenarios

Assign prefix to Docker for use by containers

```
docker –d --ipv6 –fixed-cidr-v6="2001:db8:1::/64"
```

Configure proxy for NDP

```
sysctl net.ipv6.conf.eth0.proxy_ndp=1
ip -6 neigh add proxy 2001:db8::abcd dev eth0
```

- IPv4 –> IPv6 flows for containers desirable

- Highly orchestrated create/destroy lifecycle
- Only some containers require public communication
- SIIT-DC suitable solution (draft-ietf-v6ops-siit-dc-03)

- Add dynamic creation of SIIT-DC entries by
  - DNS lookup for AAAA record referring to container
  - SIIT-DC translation entry removed when container destroyed or at TTL of AAAA
- Or
  - Flagged as required in container configuration
  - SIIT-DC translation entry removed when container destroyed
  - Needs cost model otherwise every container owner sets “public” flag “just in case”