

- Minikube start
 - Downloads an iso file
 - Talks to VB and bring up a vm

VM Minikube: 192.168.99.100

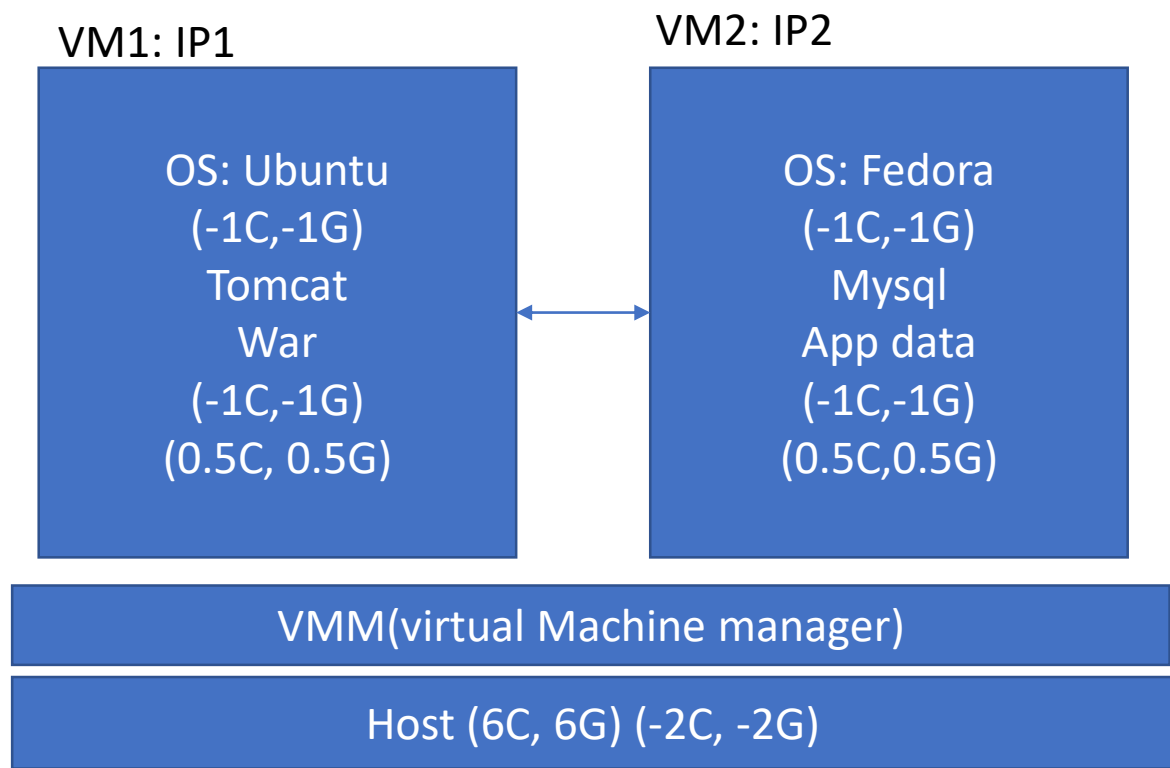
docker

k8s

Virtual box(VMM)

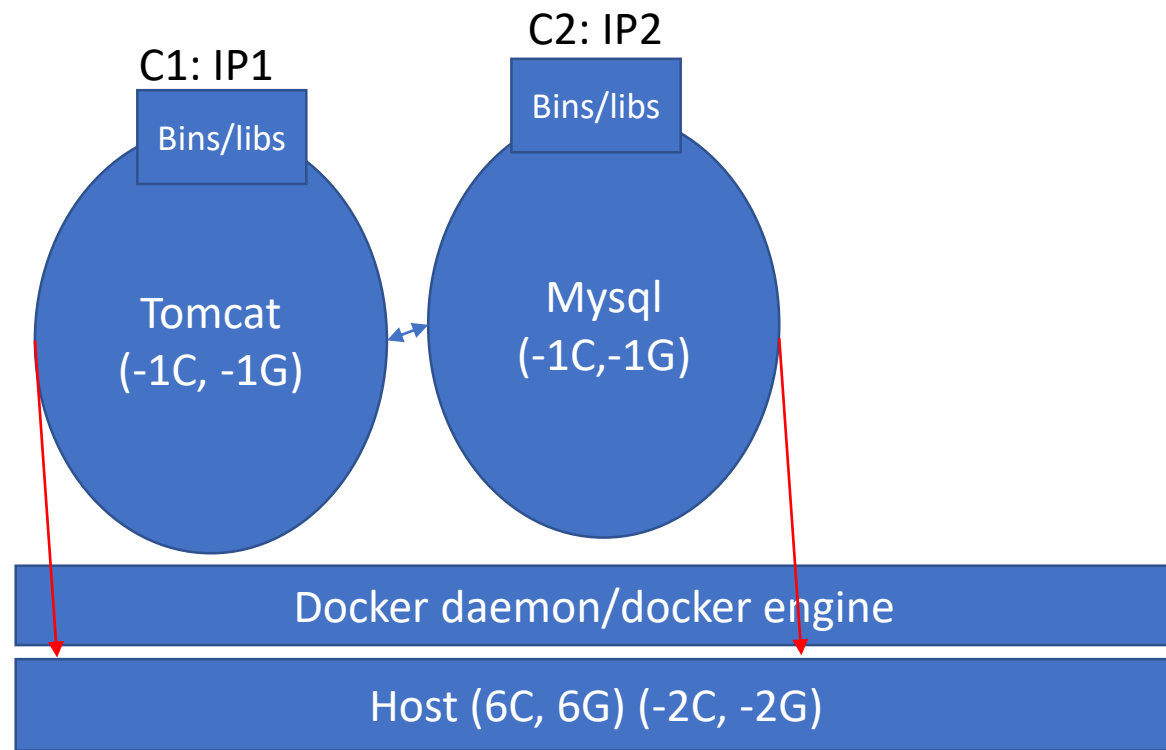
Windows/macOS

1. speed: mins
2. Share-ability: snapshot/checkpoint/appliance: 1.6 GB
3. Repeat-ability: run sheets, scripts
4. Bang-for-buck: limited, fragmented

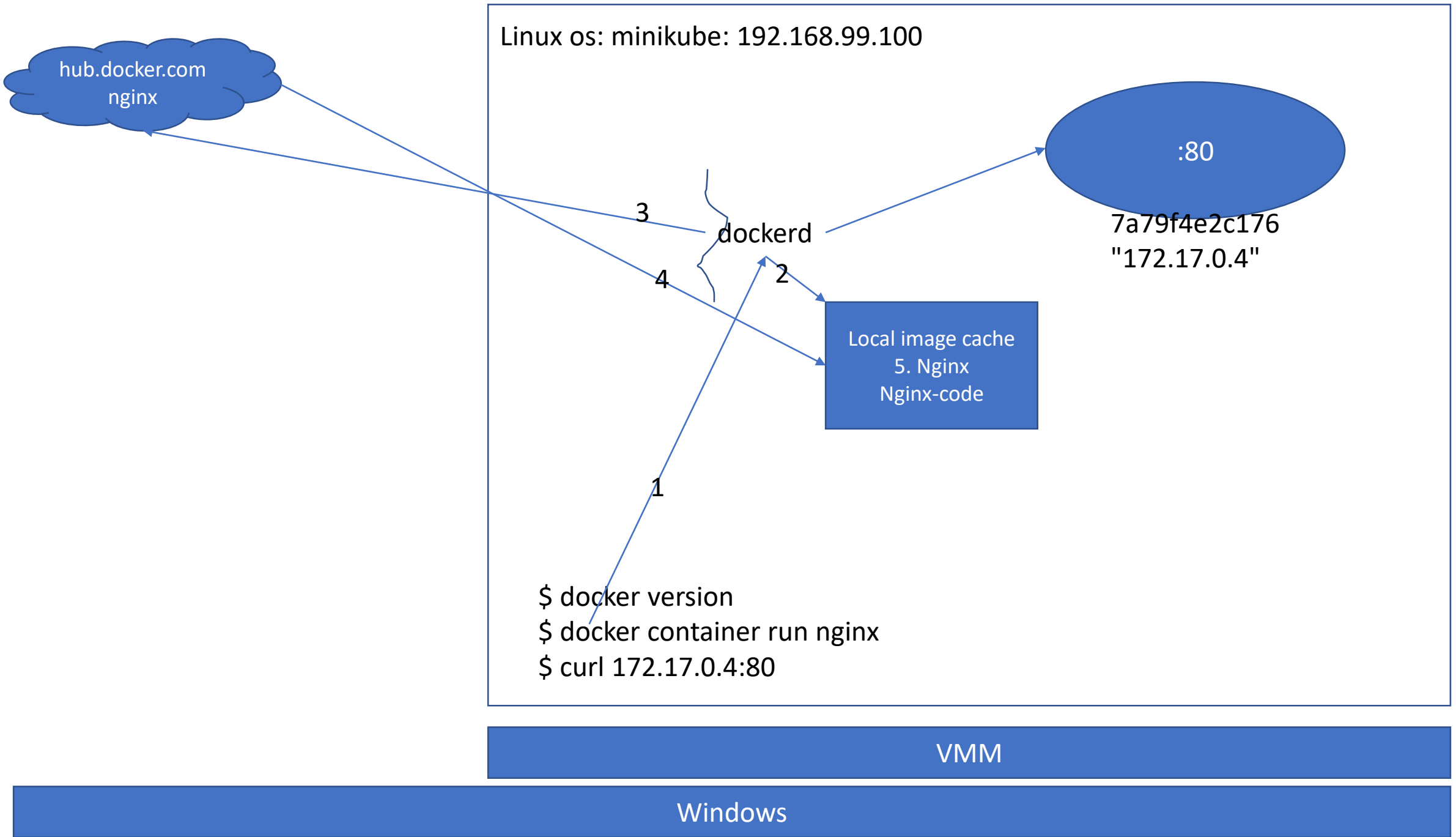


Isolation: Network, file system, process

1. speed: ms
2. Share-ability: images
3. Repeat-ability: DSL, code
4. Bang-for-buck: ++, not fragmented



Isolation: Network, file system, process



Windows/host os

Virtualhostonly network
adapter #4
Low: 192.168.99.100
Upper: 192.168.99.255
Gateway: 192.168.99.1

- Ping 192.168.99.100 -- will work
- Ping 172.17.0.4 -- cannot work

http://192.168.99.100: 9999

Minikube: 192.168.99.100

Docker0/bridge
Gateway: 172.17.0.1

Nginx:80

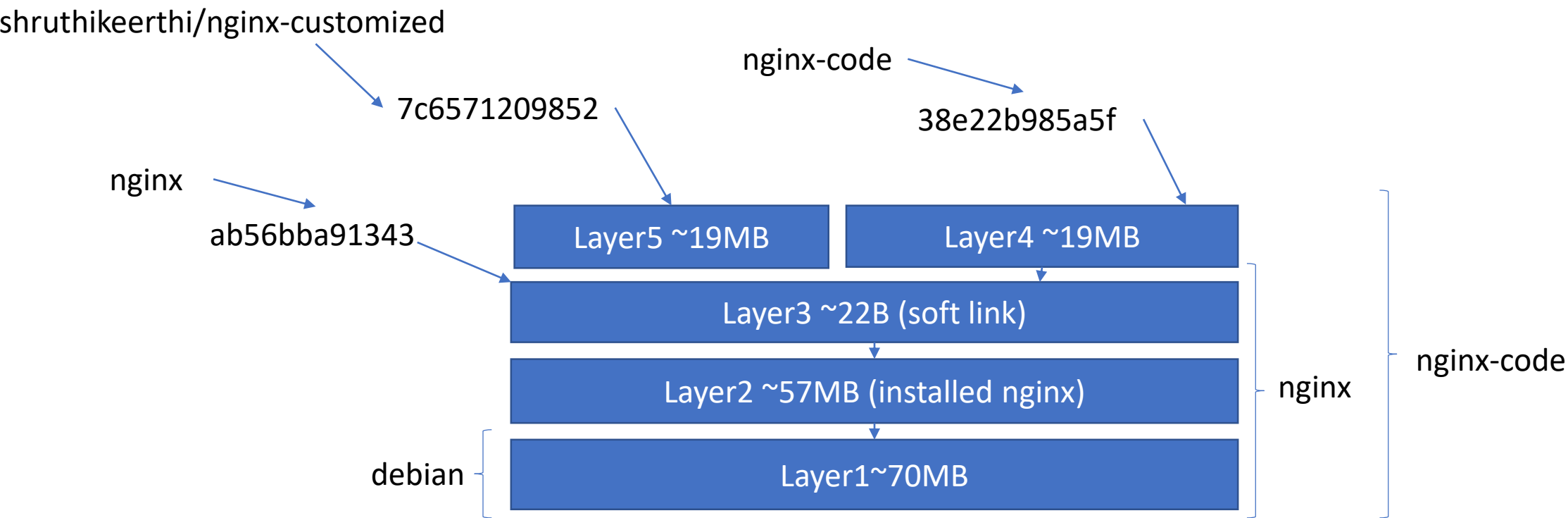
172.17.0.4:80

\$ curl 172.17.0.4

Port-fwd/port-publish

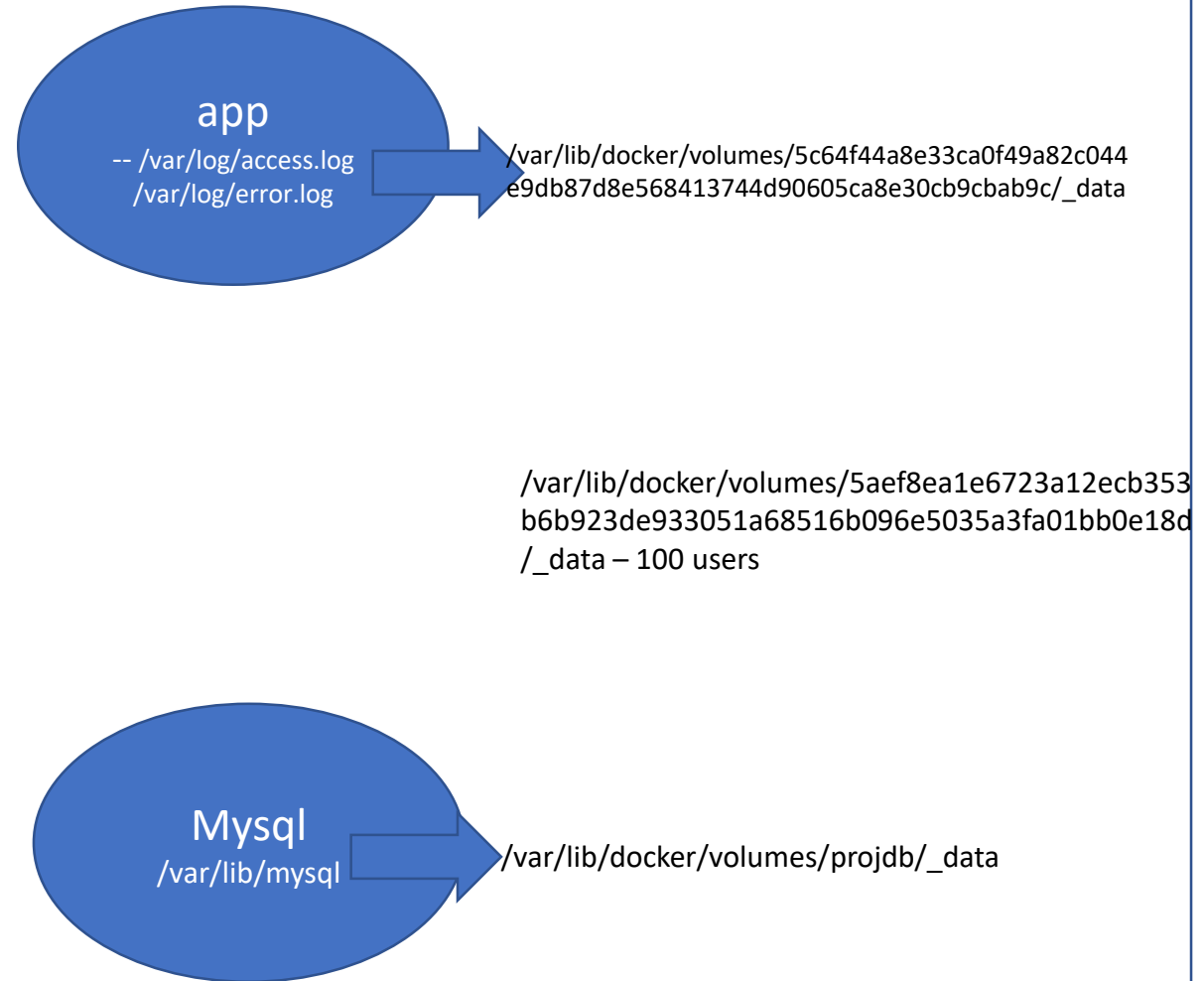
9999

Image: Unified File System

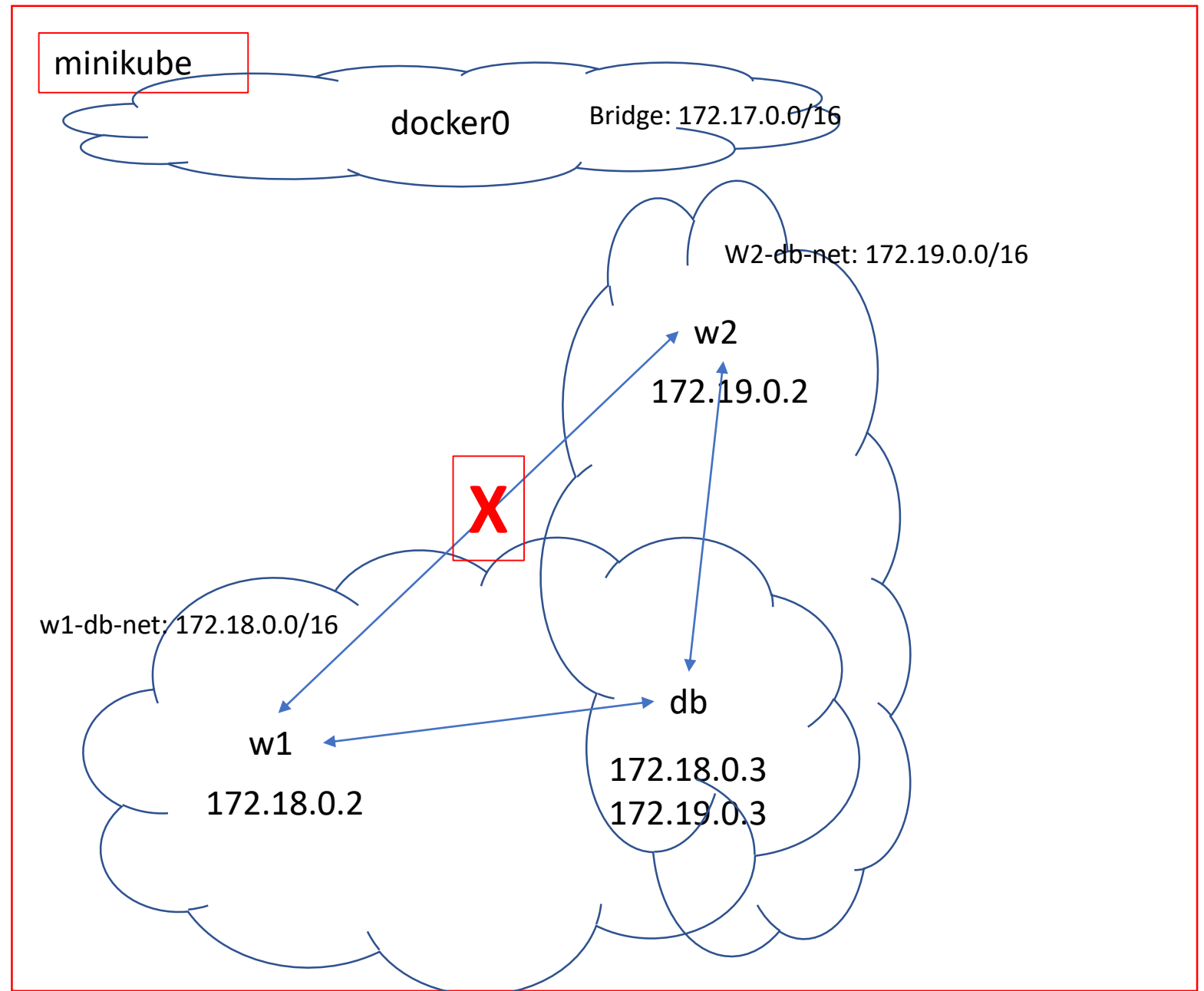


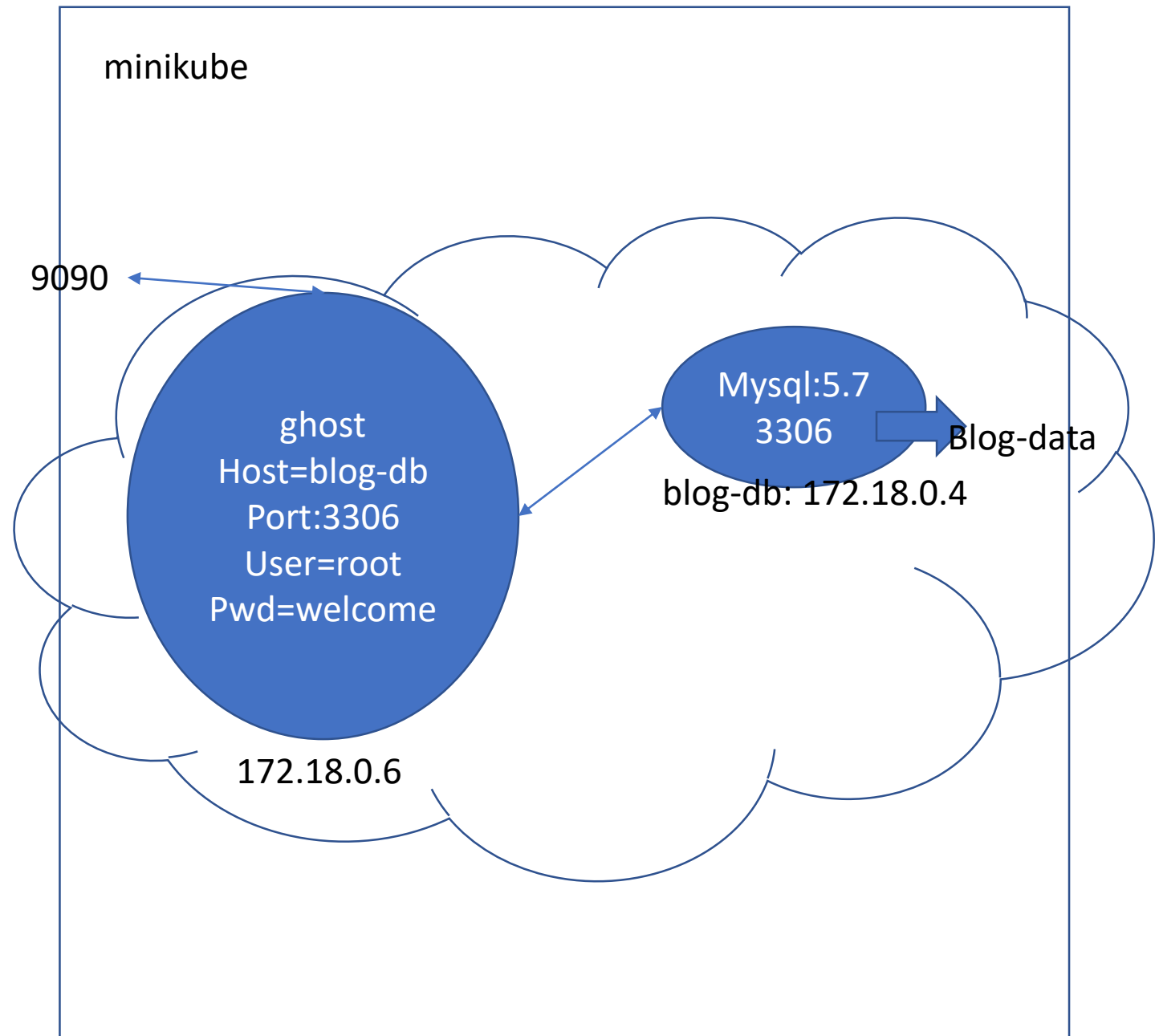
Containers fS : ephemeral

minikube



1. ipaddress → dns service
2. Better isolation: w1 → db, w2 → db,
w1 – X – w2





Container Orchestration

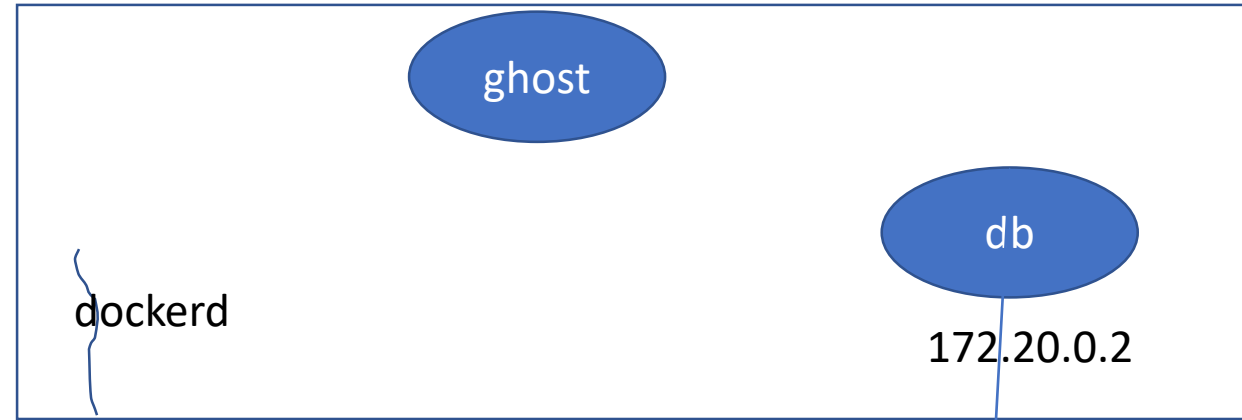
1. Application scaling
2. Load balancing
3. Interact and coordinate between multiple dockerd
4. Networking
 1. Ip address management
 2. No node awareness
5. Dns services
6. Node (VM, on prem, on cloud, physical) management -> cluster management

Products:

K8s

Docker swarm

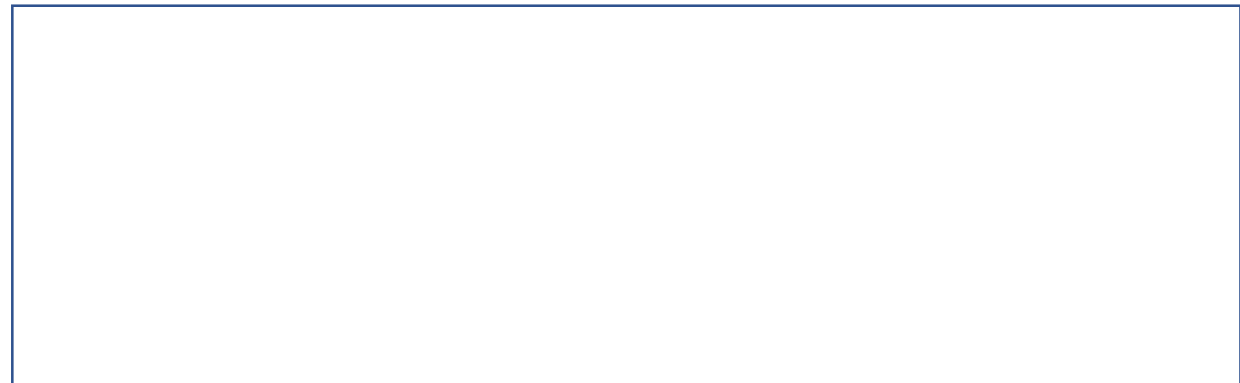
Node1 8c, 124g



Node2 8c, 124g



Node3 8c, 124g

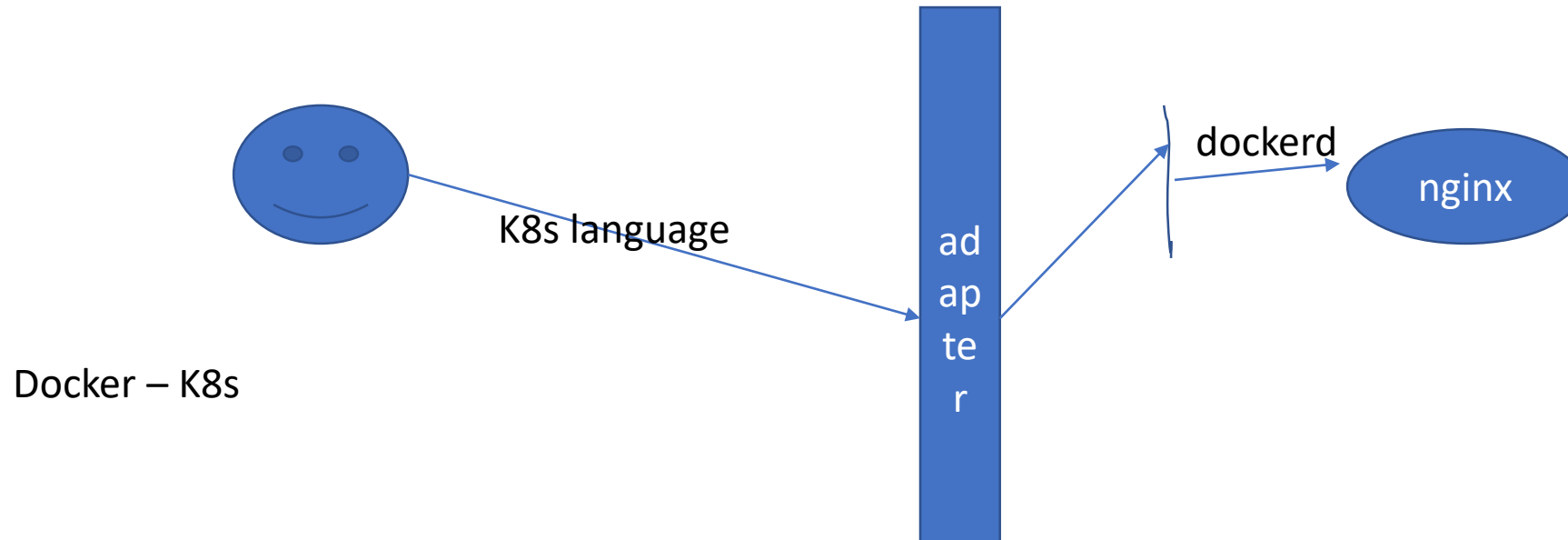


Docker Swarm

1. Open source, Docker
2. Native support for docker containers
3. Learning curve is smooth
4. DS → docker
5. Vendor-lockin
6. Fairly new

K8s

1. Open source, Google
2. Terminology or adapters
3. Learning curve is a litter steeper
4. K8s -> any container provider which implement Open Container Spec
5. No vendor lockin
6. Battle tested → benchmarked against 10000's nodes



Learning K8s

1. Terms and concepts k8s – single node cluster
2. Communication – single node clust
3. Architecture
4. Multi-node cluster k8s

minikube

K8s-service
(pod-def)

dockerd

Pod cont

"172.17.0.4"

Shared-
veth

Web:ngin
x:80

Pod: 172.17.0.4

Web:ngin
x:80

\$ curl <pod's ip>:<port of cont>

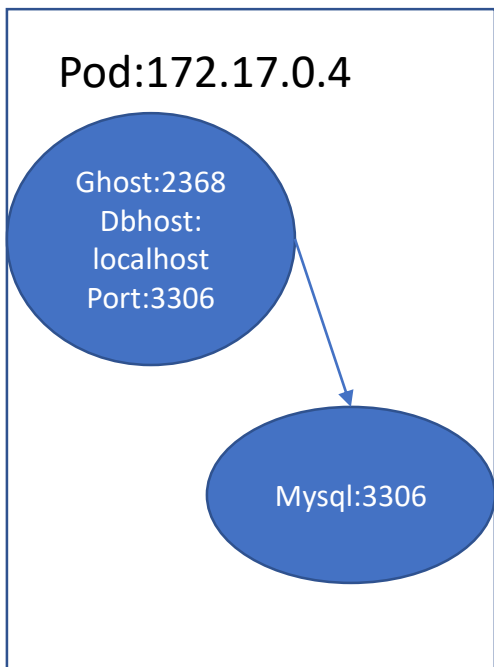
\$ curl 172.17.0.4:80

➤ Kubectl (refers to ~/.kube/config)

Server version

➤ Kubectl apply -f pod.yml

1. pod-cont provides n/w access to containers
2. Make sure that always the required containers run
3. Pod can have multiple containers provided they come up on different port number
4. Design time decision: Should I put containers in the same pod --- if they scalable together



Deployment not scalable
as it creates copies of DB
too

