

Kubernetes

for Database Administrators



Database





Host machine



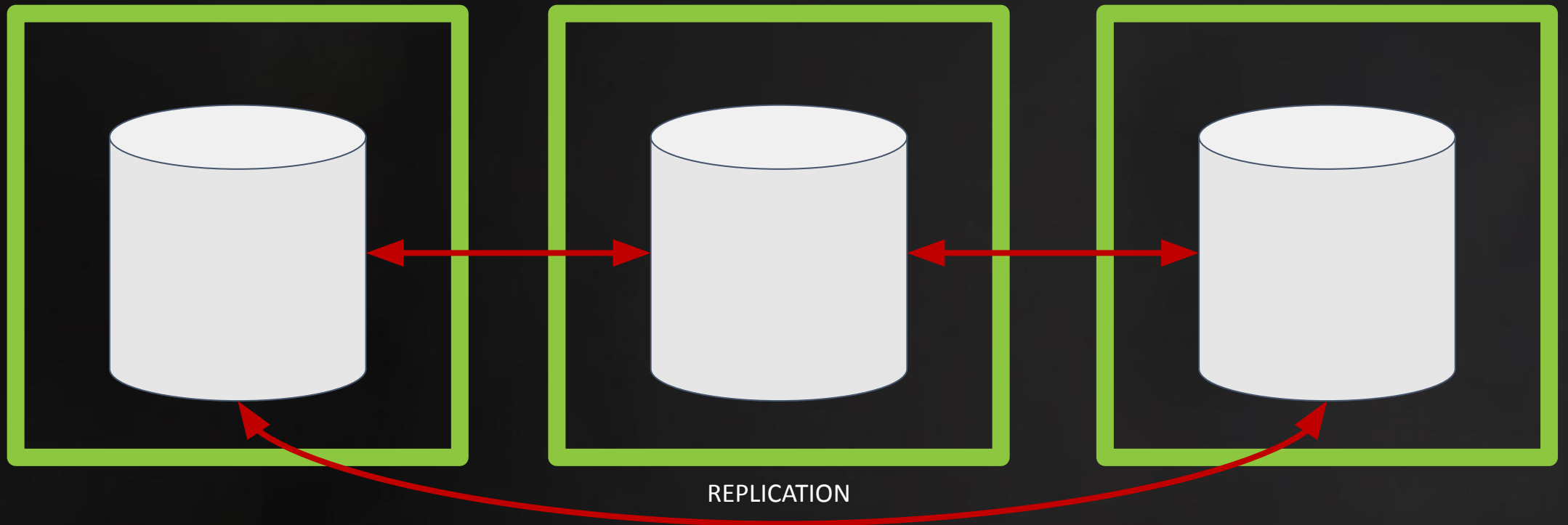


Host machine
What happens when
this goes down?

Let's add more
machines!

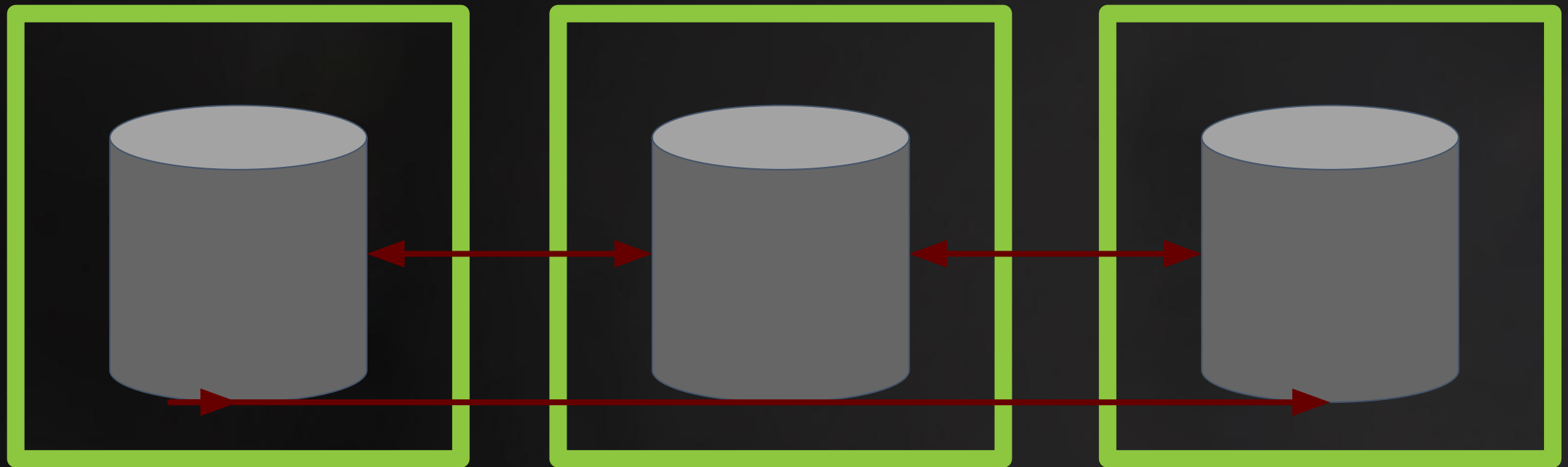


Let's add more
machines!



Let's add more
machines!

What happens when
these go down?



DBA

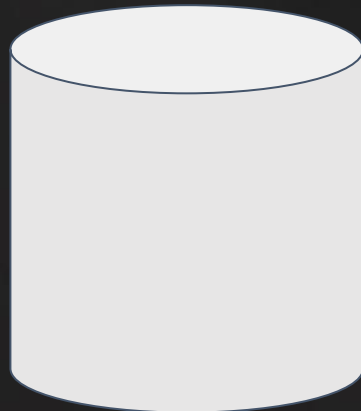
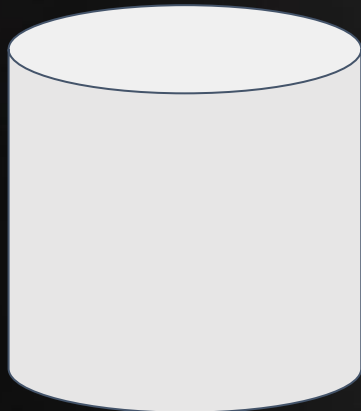
THE #1 ~~PROGRAMMER~~ EXCUSE
FOR LEGITIMATELY SLACKING OFF:

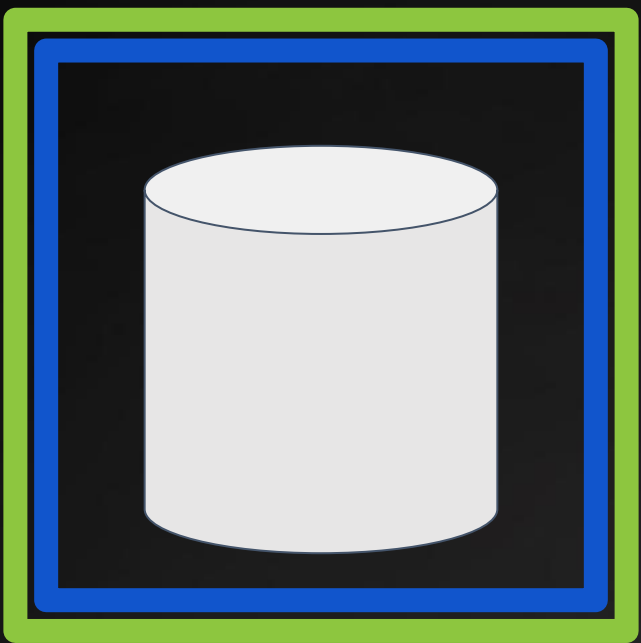
~~"MY CODE'S COMPILING."~~
MY DB IS INSTALLING.



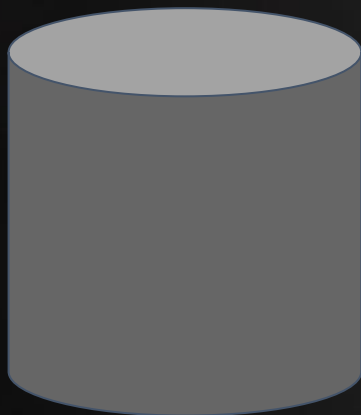


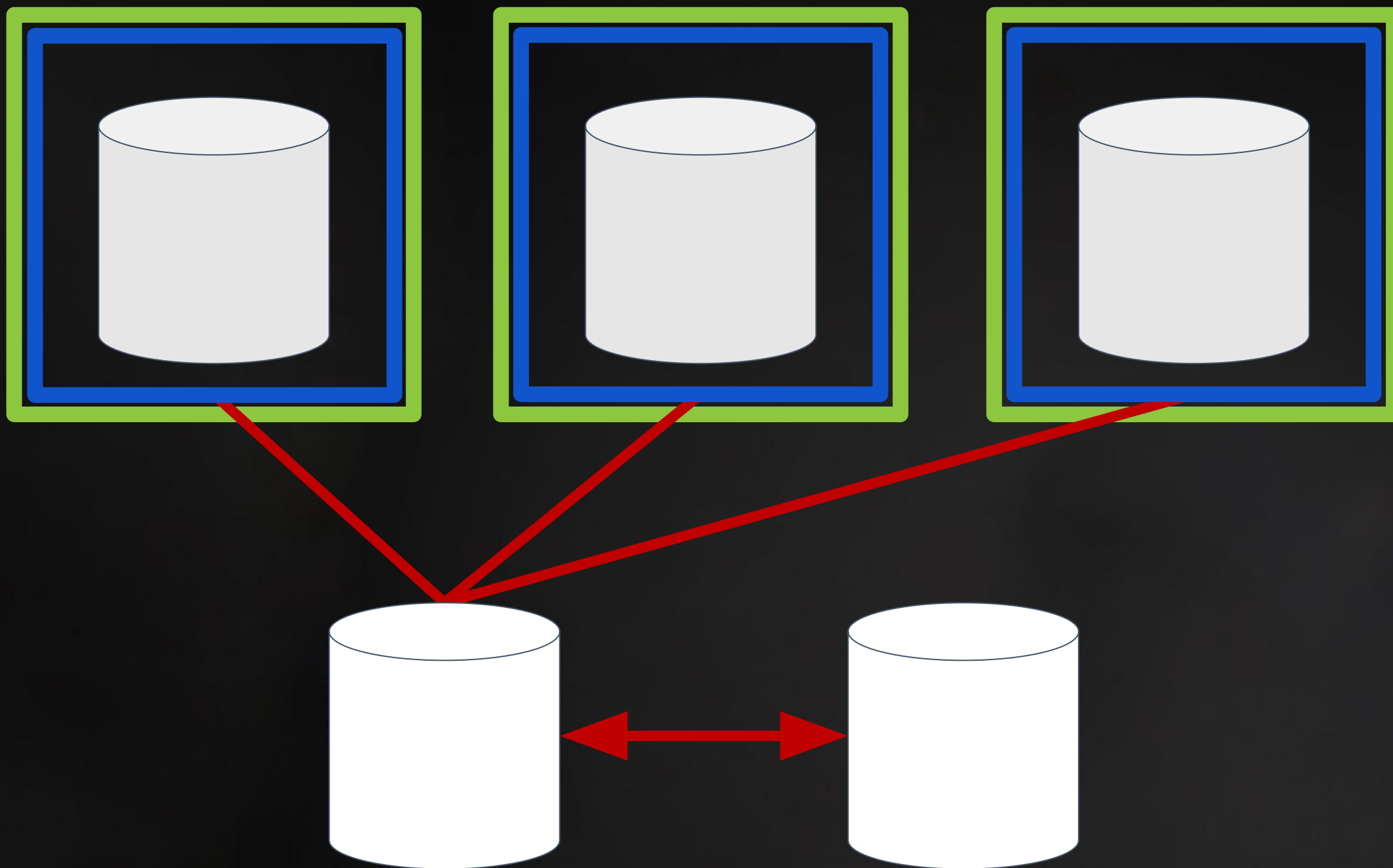
Very Expensive™
Enterprise™ Storage®

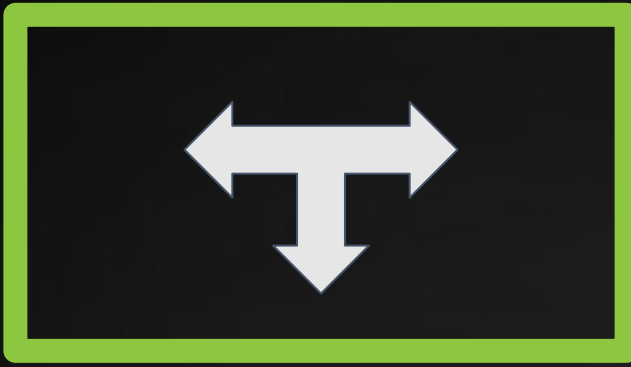




Very Expensive™
Enterprise™
Virtualization®

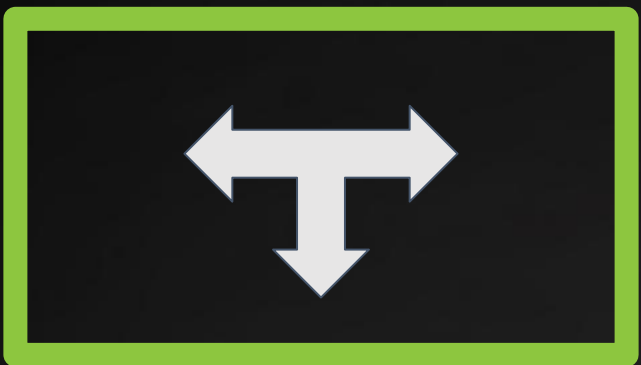






Very Expensive™
Enterprise™
Load Balancer®



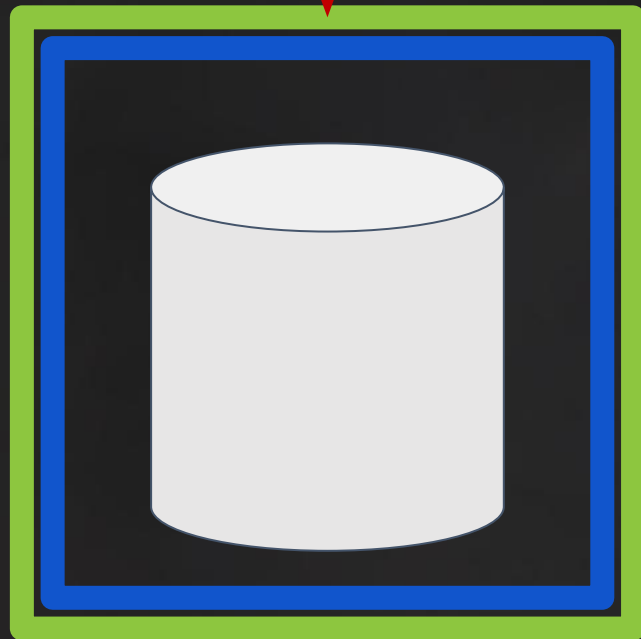


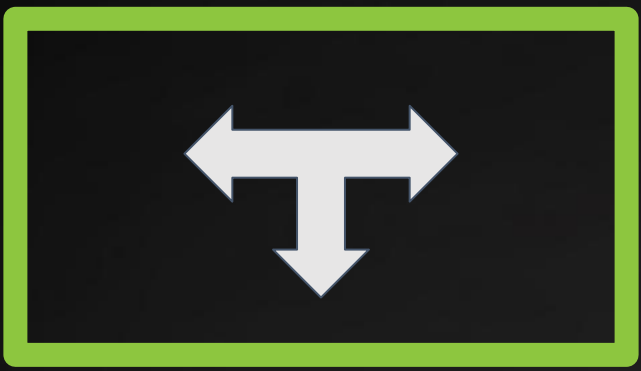
Take this out of
the LB first



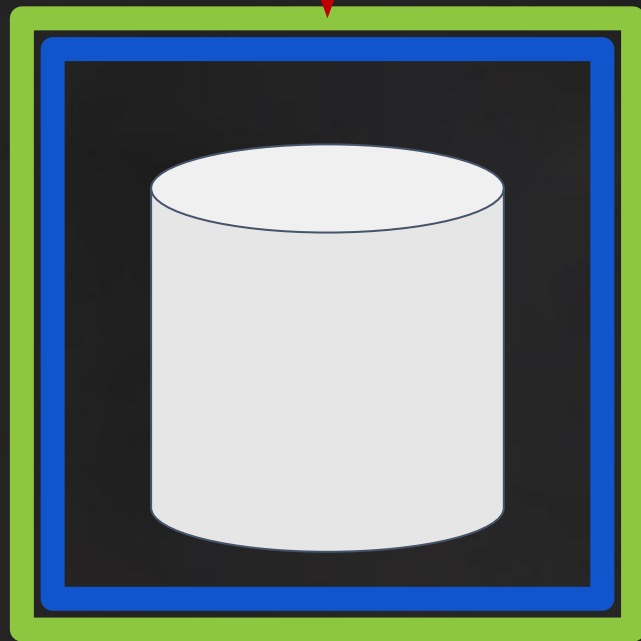


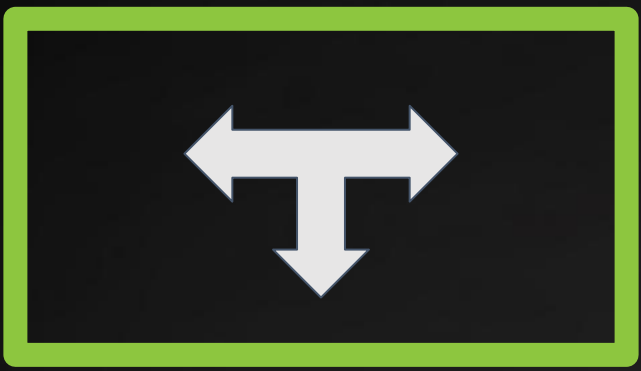
Upgrade
this





Put it back



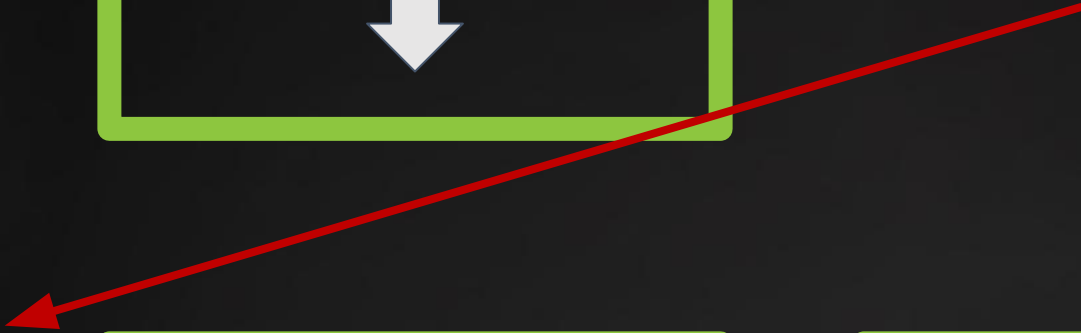


Then
this

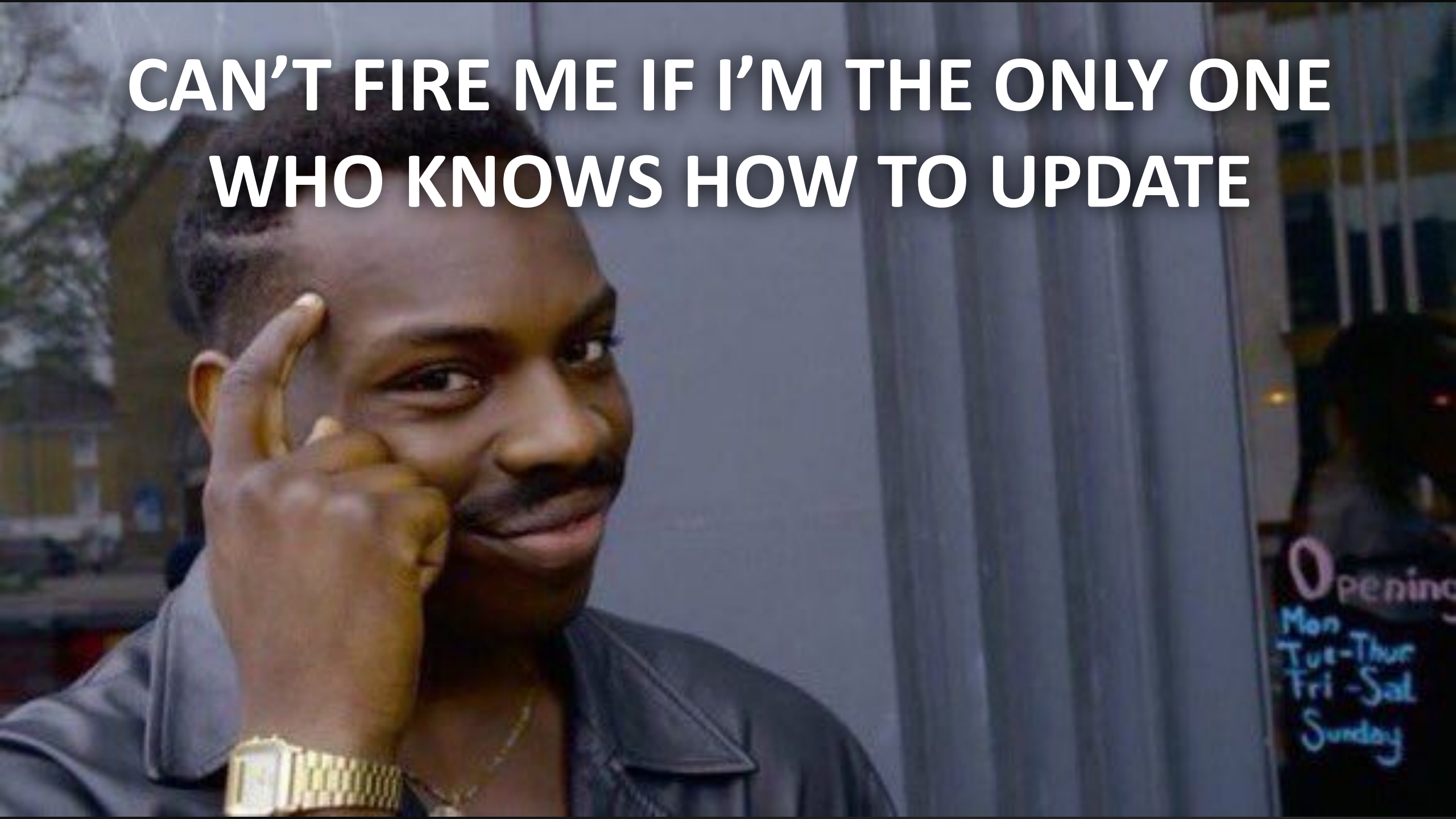




Then
this

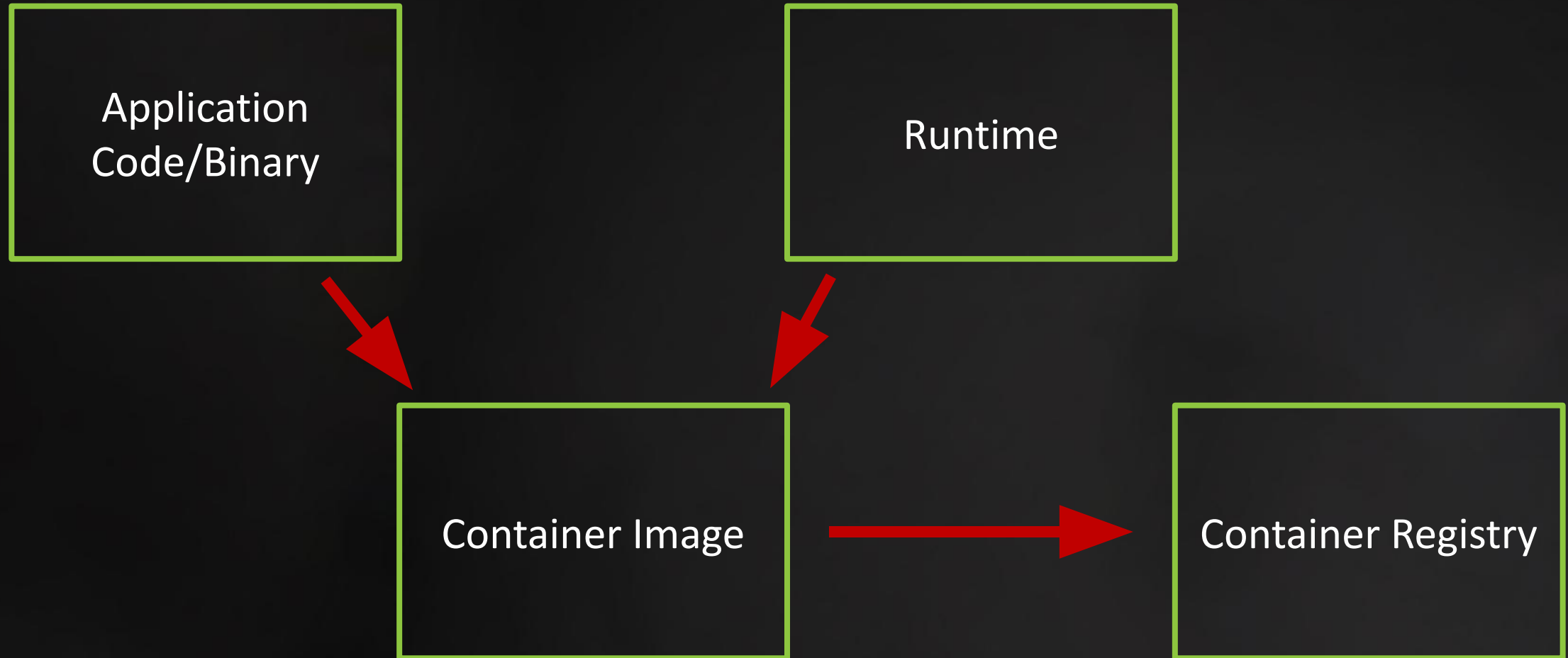


**CAN'T FIRE ME IF I'M THE ONLY ONE
WHO KNOWS HOW TO UPDATE**



Containers

Container Image Build Process



```
FROM alpine
RUN apk upgrade --no-cache && \
    apk add --no-cache 'libssl1.1>=1.1.1k-r0' 'libcrypto1.1>=1.1.1k-r0'
COPY ./containerssh/containersssh-testauthconfigserver /
COPY ./containerssh/LICENSE.md /
COPY ./containerssh/NOTICE.md /
RUN chmod +x /containersssh-testauthconfigserver
ENTRYPOINT ["/containersssh-testauthconfigserver"]
CMD []
USER 1080:1080
EXPOSE 8080
```

```
FROM alpine
```

```
RUN apk upgrade --no-cache && \  
    apk add --no-cache 'libssl1.1>=1.1.1k-r0' 'libcrypto1.1>=1.1.1k-r0'  
COPY ./containerssh/containersssh-testauthconfigserver /  
COPY ./containerssh/LICENSE.md /  
COPY ./containerssh/NOTICE.md /  
RUN chmod +x /containersssh-testauthconfigserver  
ENTRYPOINT ["/containersssh-testauthconfigserver"]  
CMD []  
USER 1080:1080  
EXPOSE 8080
```

```
FROM alpine
```

```
RUN apk upgrade --no-cache && \  
    apk add --no-cache 'libssl1.1>=1.1.1k-r0' 'libcrypto1.1>=1.1.1k-r0'
```

```
COPY ./containerssh/containersssh-testauthconfigserver /
```

```
COPY ./containerssh/LICENSE.md /
```

```
COPY ./containerssh/NOTICE.md /
```

```
RUN chmod +x /containersssh-testauthconfigserver
```

```
ENTRYPOINT ["/containersssh-testauthconfigserver"]
```

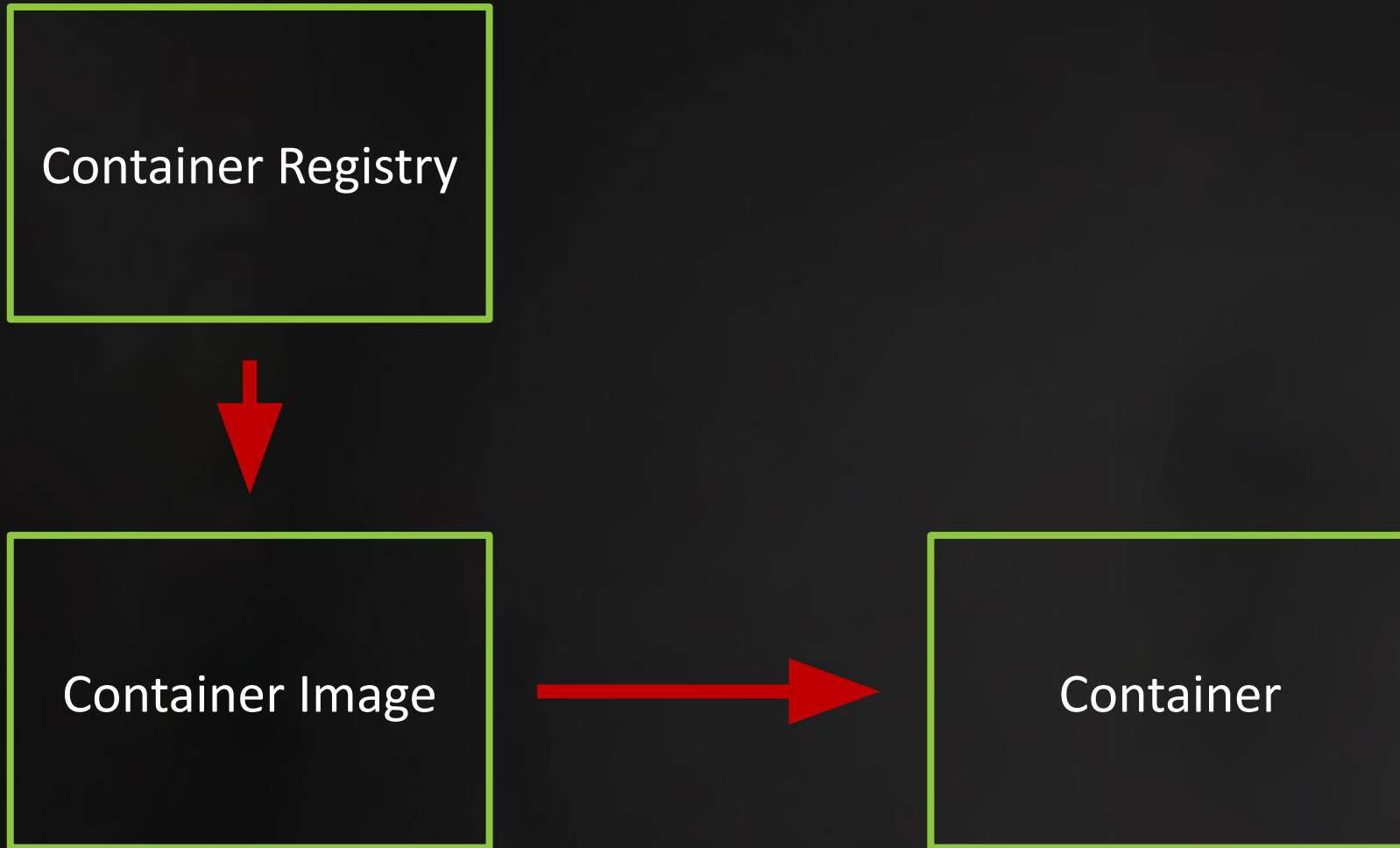
```
CMD []
```

```
USER 1080:1080
```

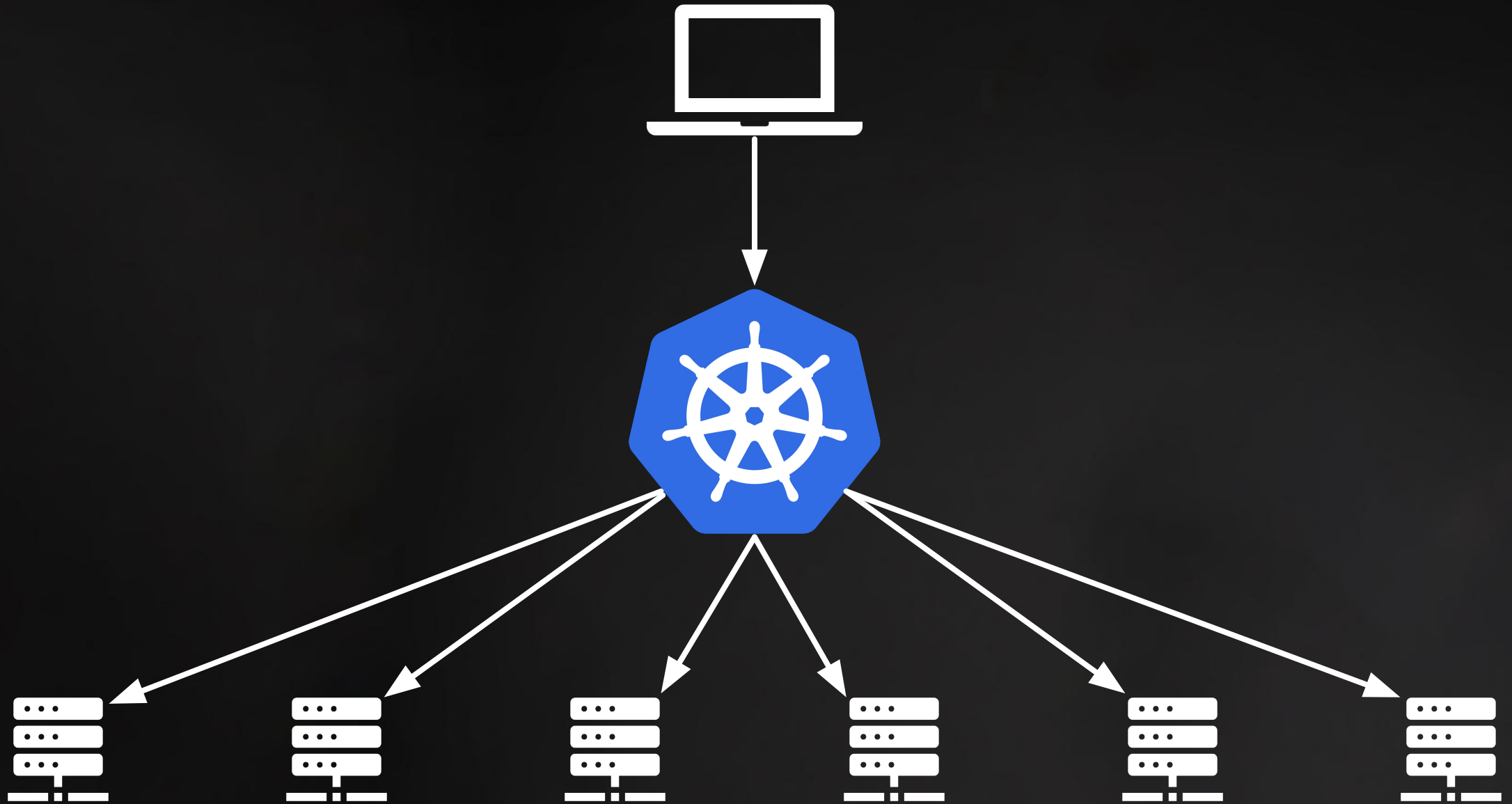
```
EXPOSE 8080
```

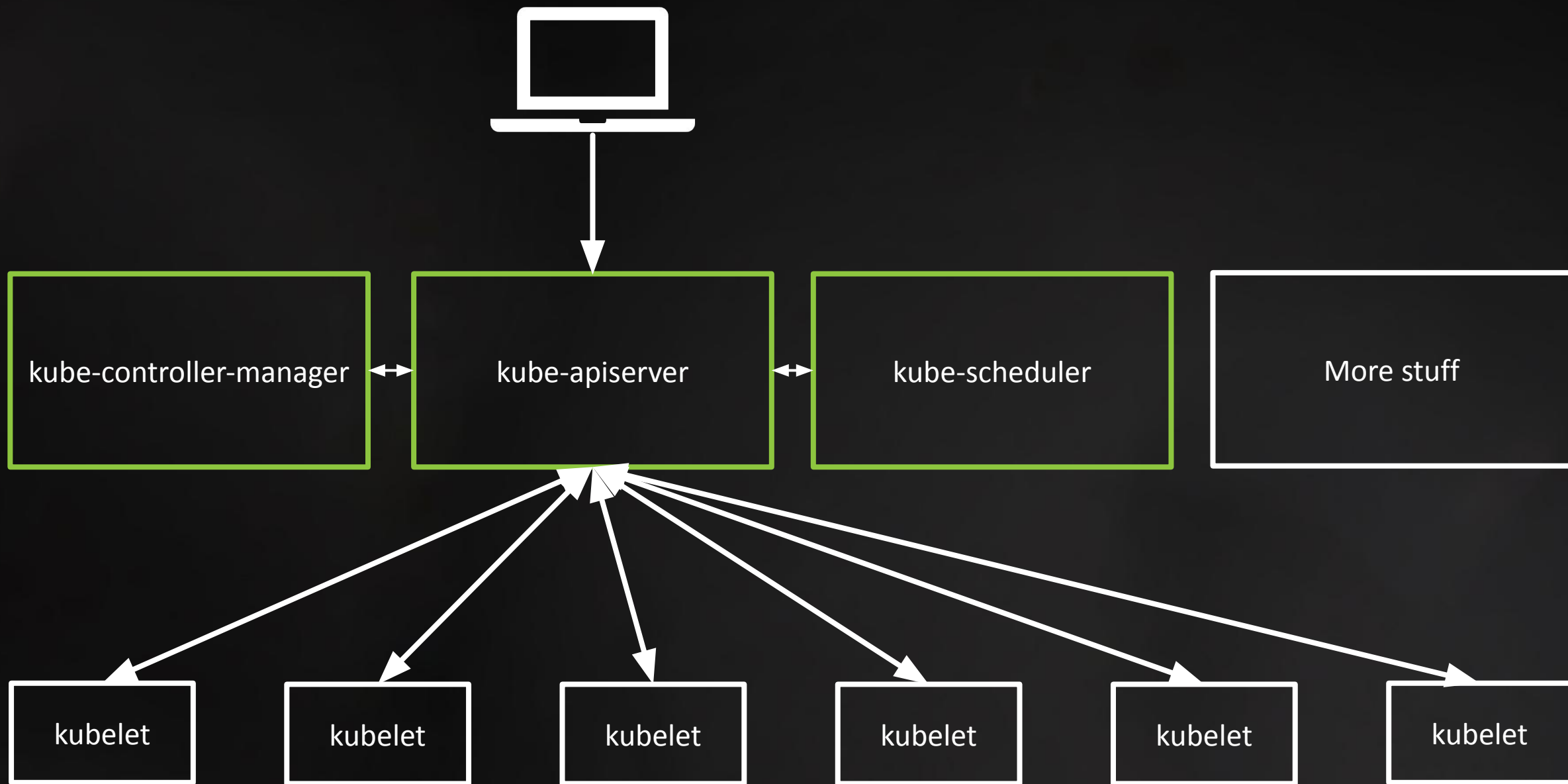
```
FROM alpine
RUN apk upgrade --no-cache && \
    apk add --no-cache 'libssl1.1>=1.1.1k-r0' 'libcrypto1.1>=1.1.1k-r0'
COPY ./containersssh/containersssh-testauthconfigserver /
COPY ./containersssh/LICENSE.md /
COPY ./containersssh/NOTICE.md /
RUN chmod +x /containersssh-testauthconfigserver
ENTRYPOINT ["/containersssh-testauthconfigserver"]
CMD []
USER 1080:1080
EXPOSE 8080
```


Running a Container









kubelet /
worker

kubelet /
worker

kubelet /
worker

kubelet /
worker

kubelet /
worker

kubelet /
worker

database

kubelet /
worker

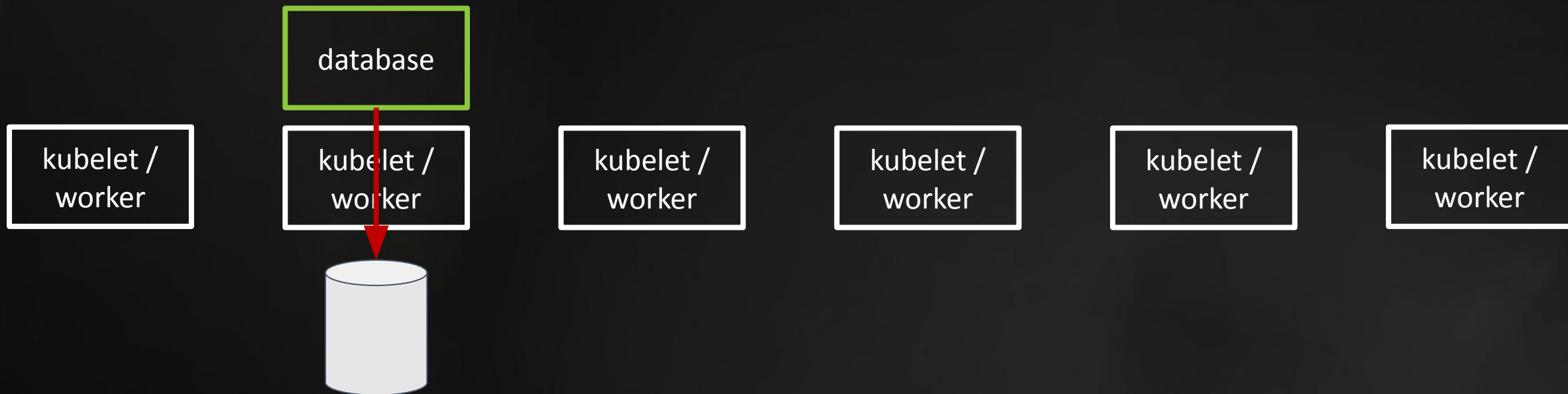
kubelet /
worker

kubelet /
worker

kubelet /
worker

kubelet /
worker

kubelet /
worker



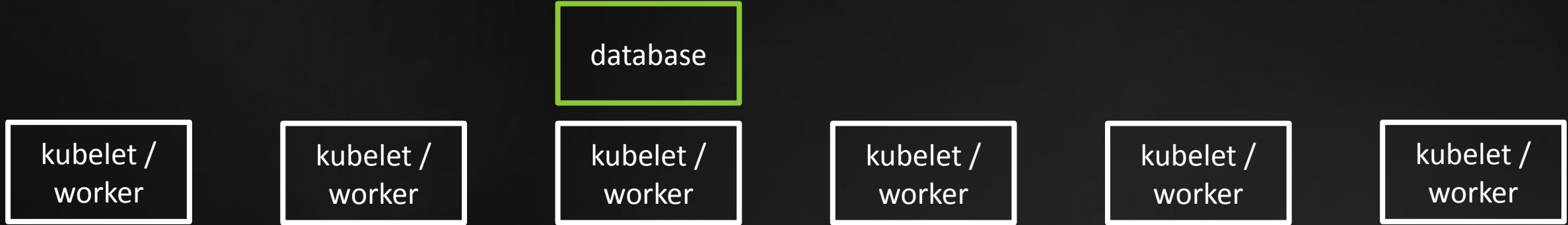
kubelet /
worker

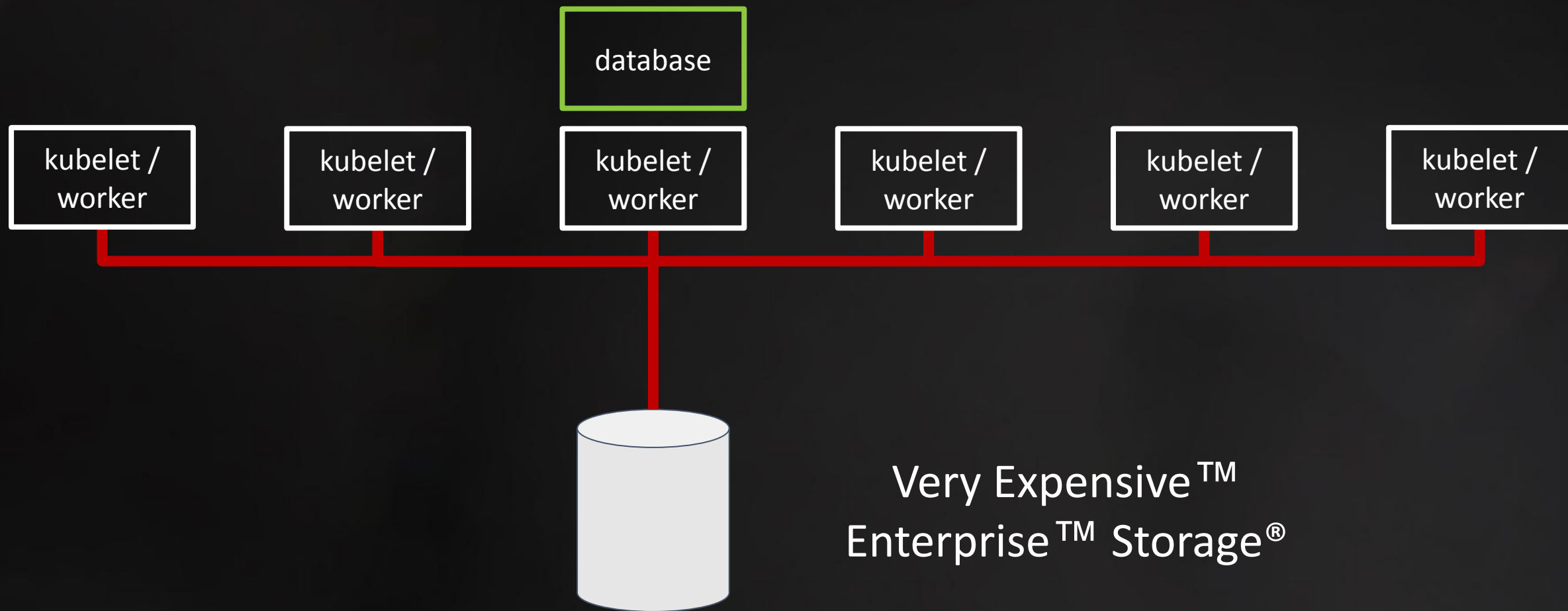
kubelet /
worker

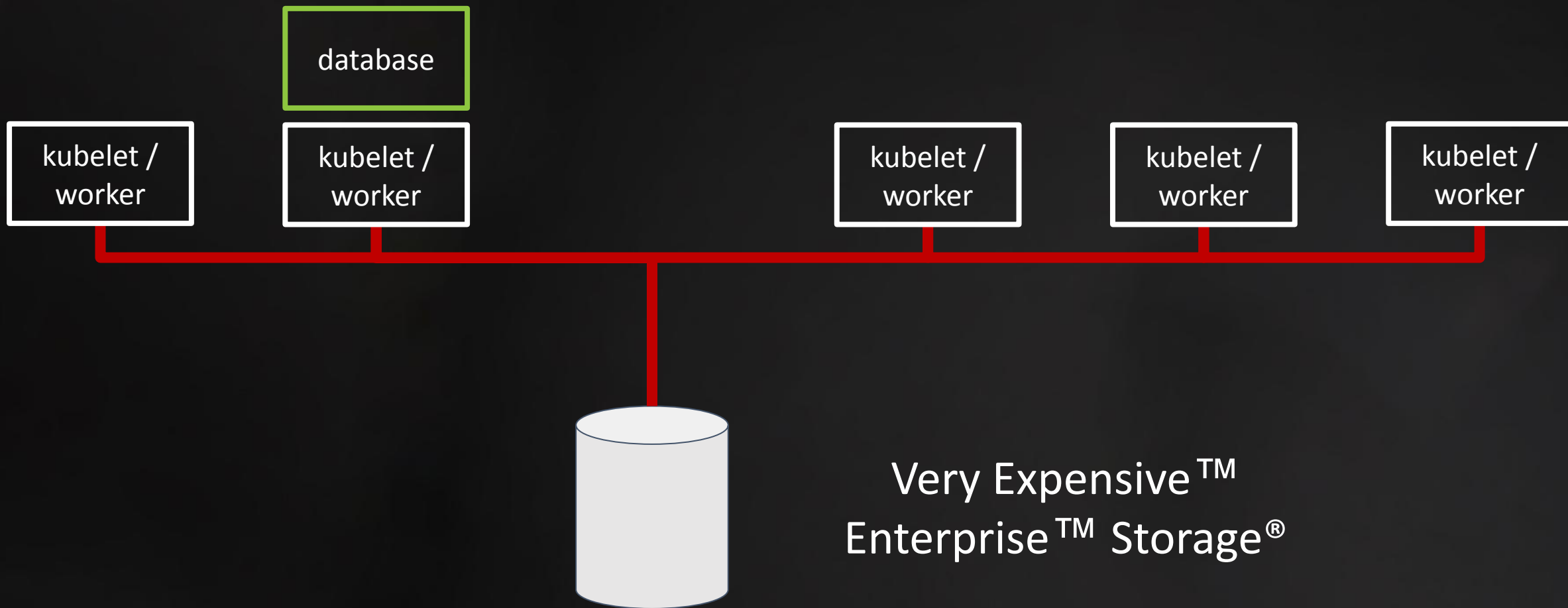
kubelet /
worker

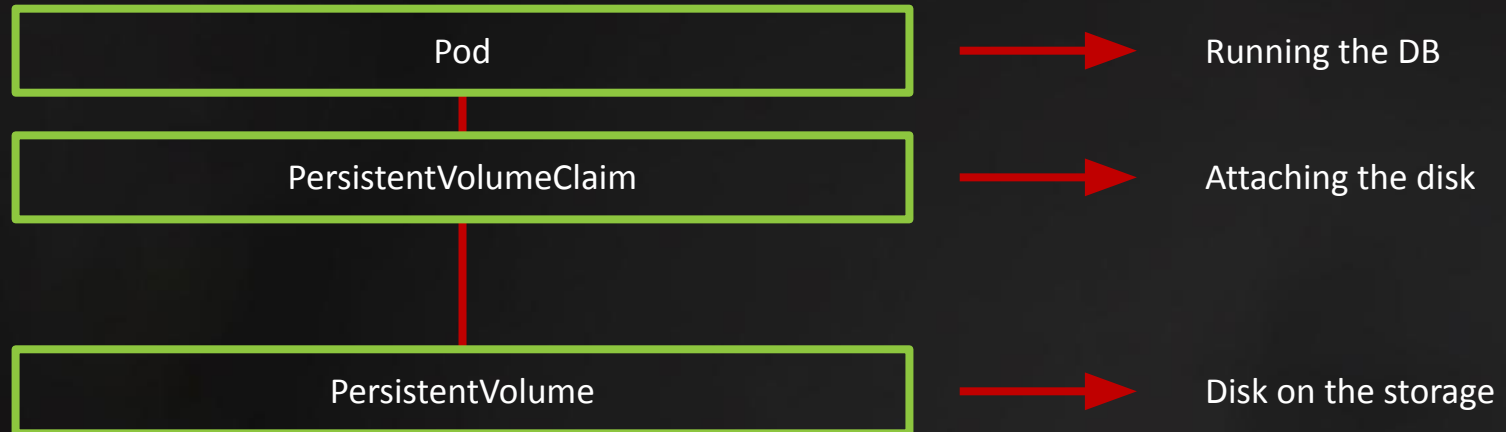
kubelet /
worker

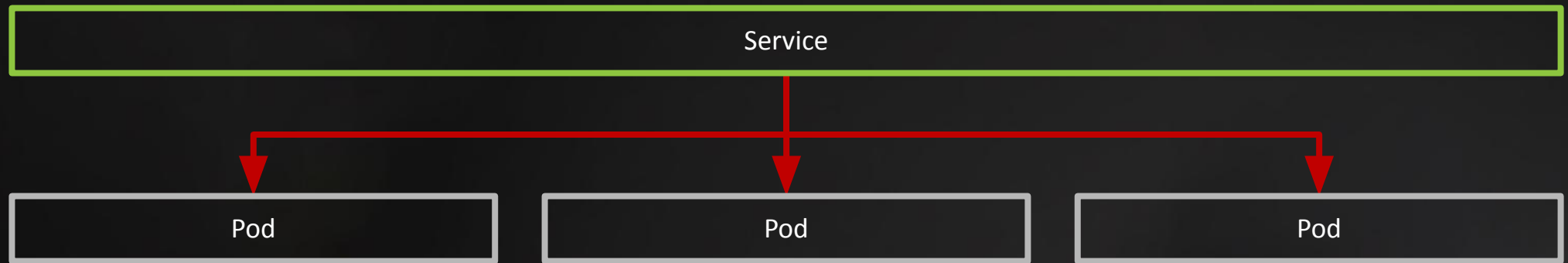
kubelet /
worker

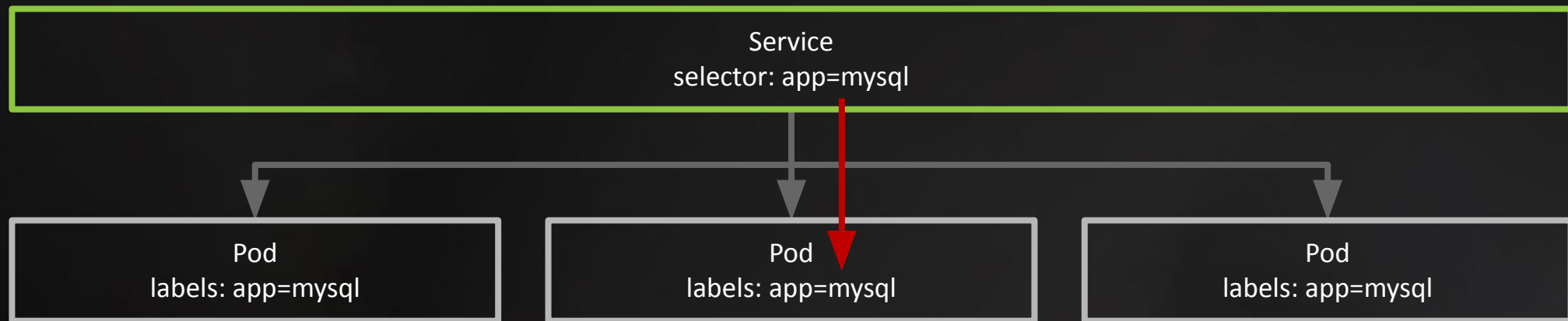








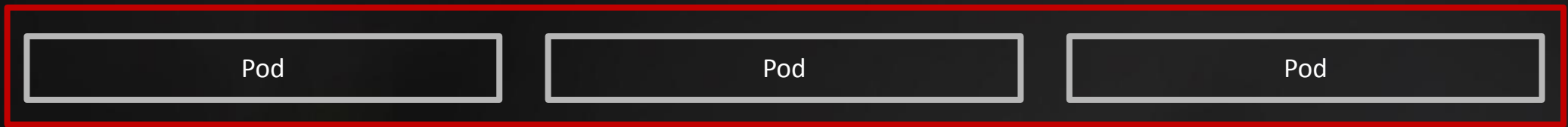




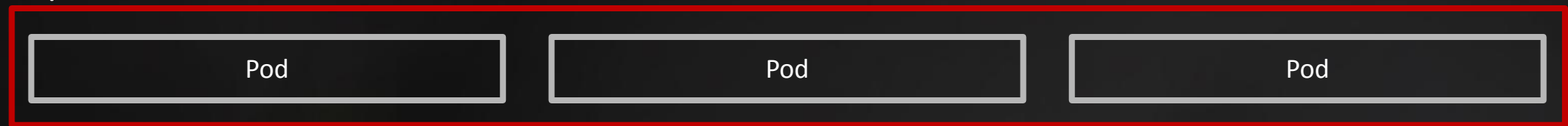
Pod

Pod

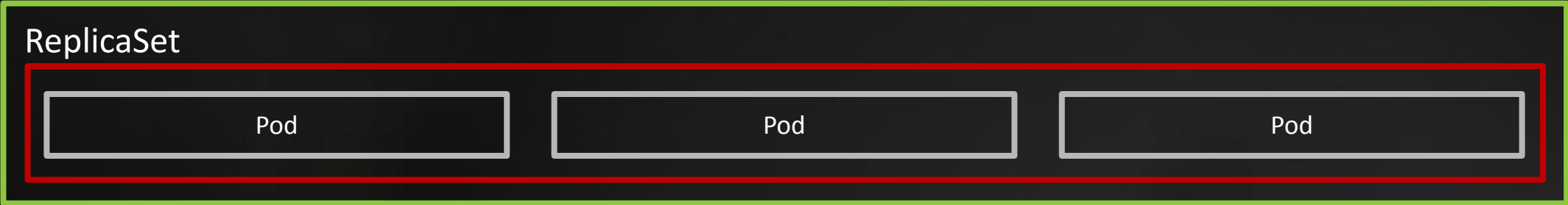
Pod



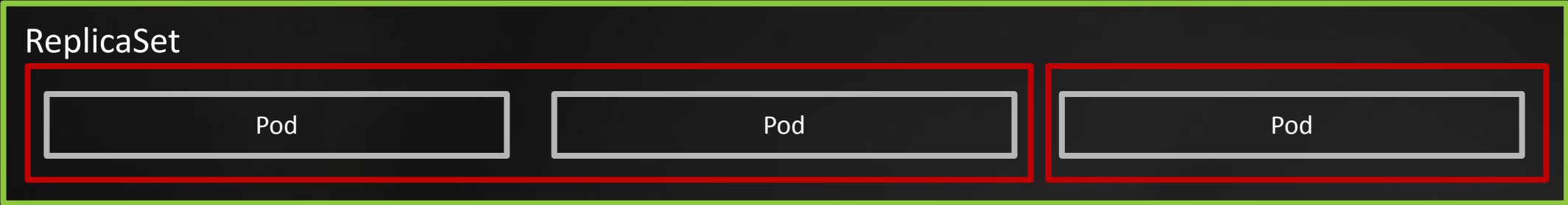
ReplicaSet



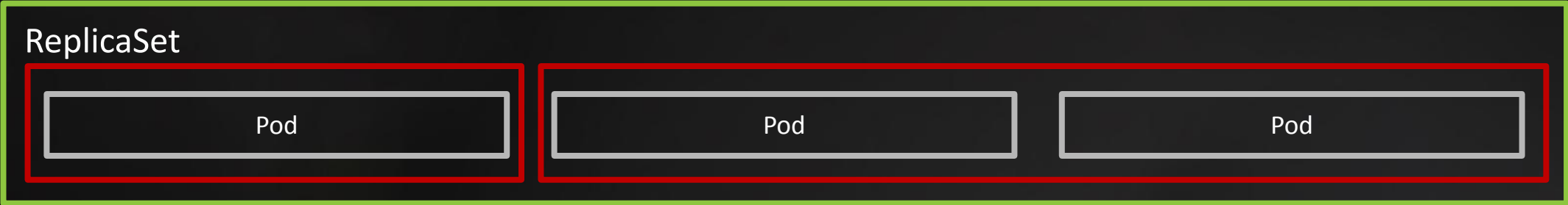
Deployment



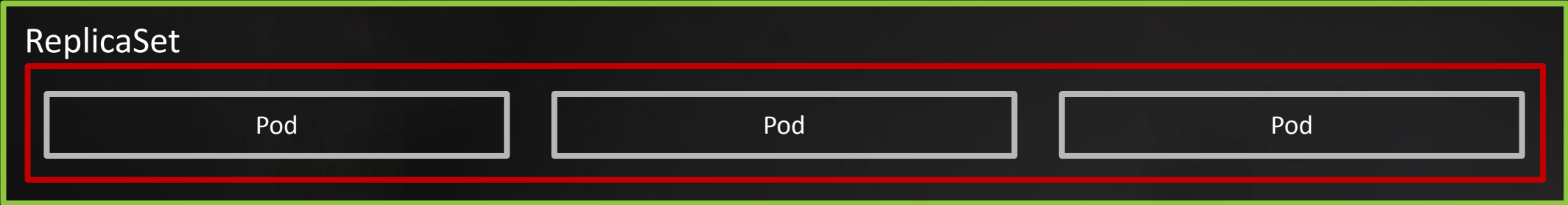
Deployment



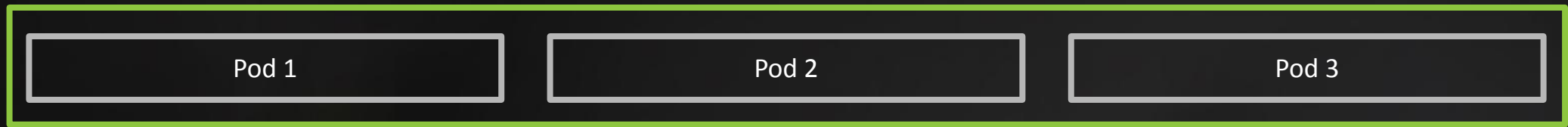
Deployment

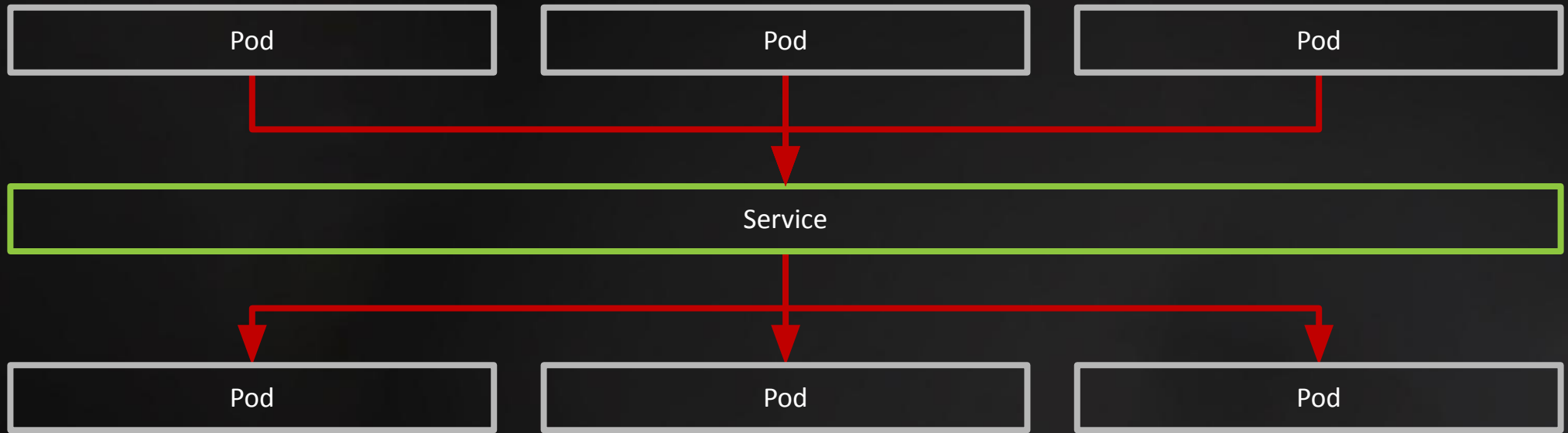


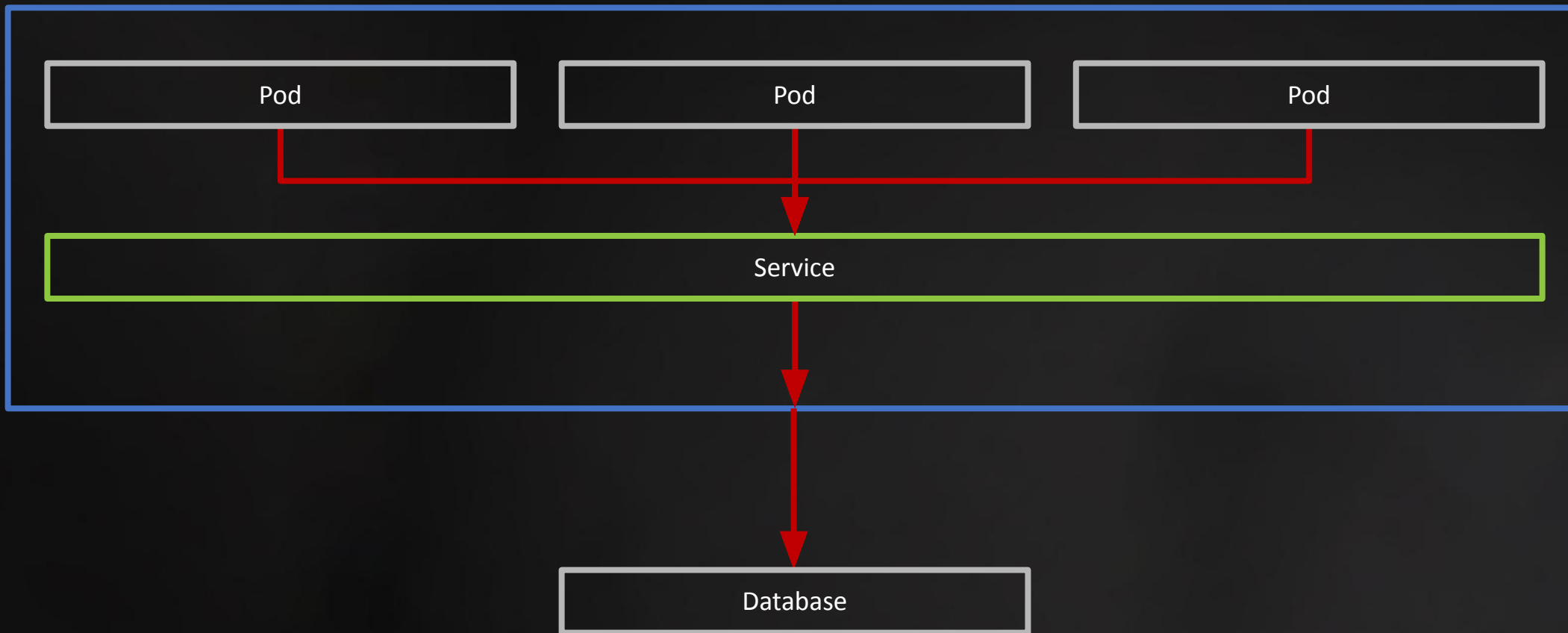
Deployment



StatefulSet

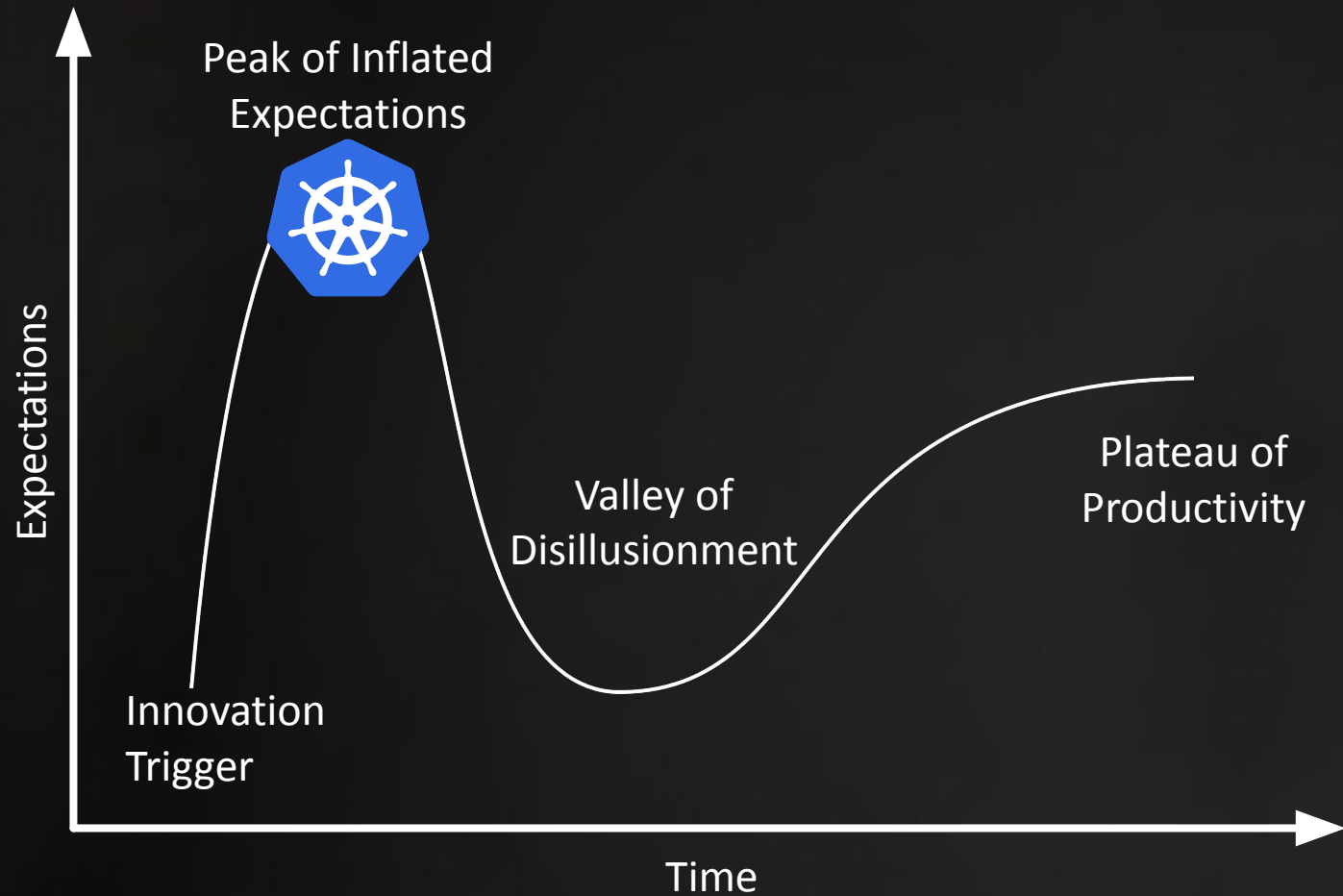


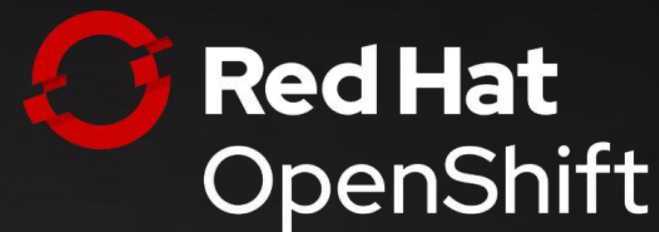





```
apiVersion: v1
kind: Service
metadata:
  name: mysql
spec:
  type: ExternalName
  externalName: debugged.it
  ports:
    - name: mysql
      port: 3306
      targetPort: 3306
```

The Gartner Hype Cycle







K3S

[DOCS](#)[GITHUB](#)

Lightweight Kubernetes

The certified Kubernetes distribution built for IoT & Edge computing

This won't take long...

```
curl -sfL https://get.k3s.io | sh -  
# Check for Ready node,  
takes maybe 30 seconds  
k3s kubectl get node
```

For detailed installation, [refer to the docs](#)

Great For

Edge

IoT

CI

ARM

```
curl -sfL https://get.k3s.io | sh -  
# Check for Ready node,  
takes maybe 30 seconds  
k3s kubectl get node
```

Thank you for listening!