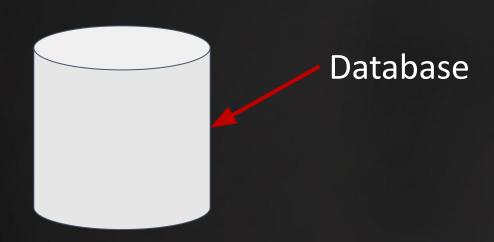
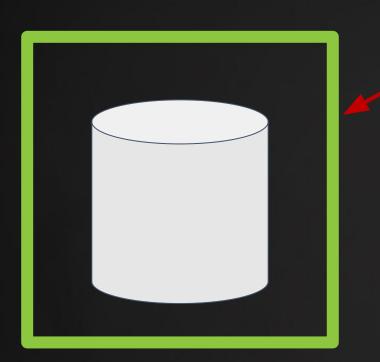
Kubernetes

for Database Administrators





Host machine

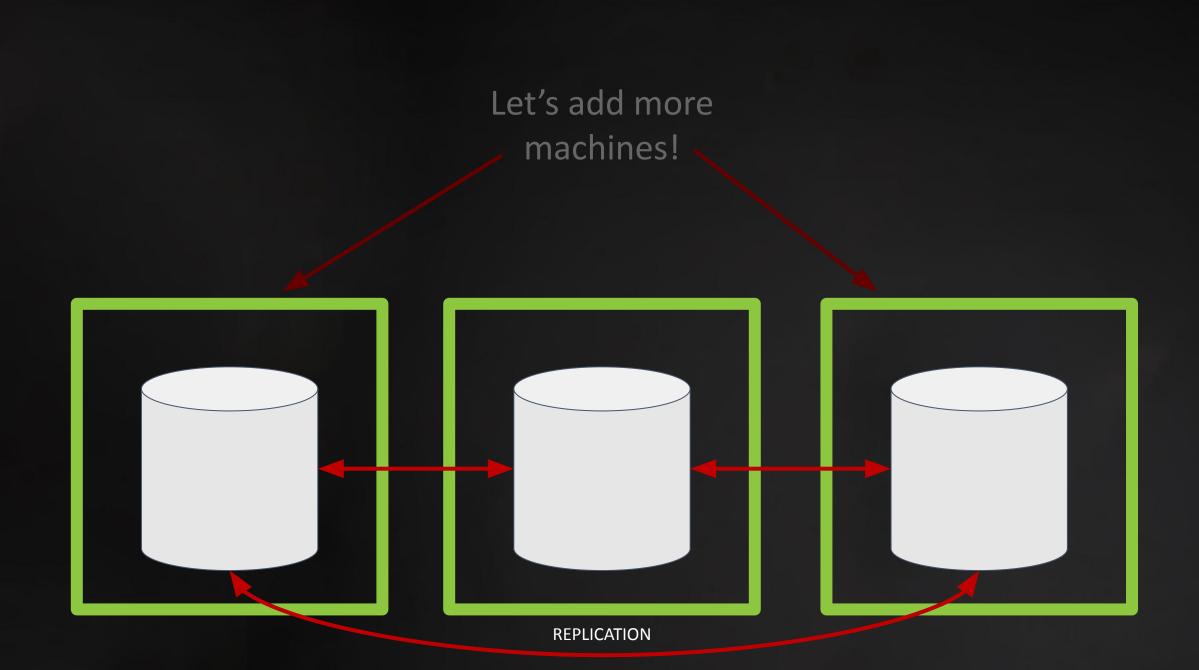


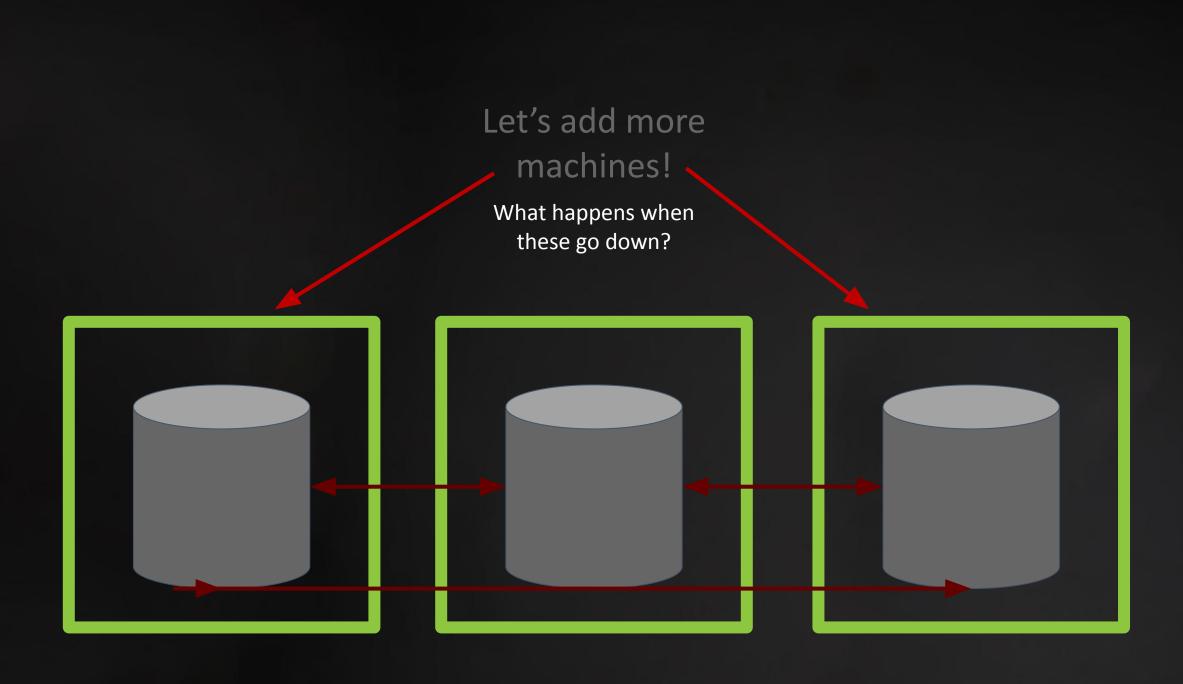
Host machine

What happens when this goes down?

Let's add more machines!





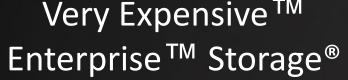


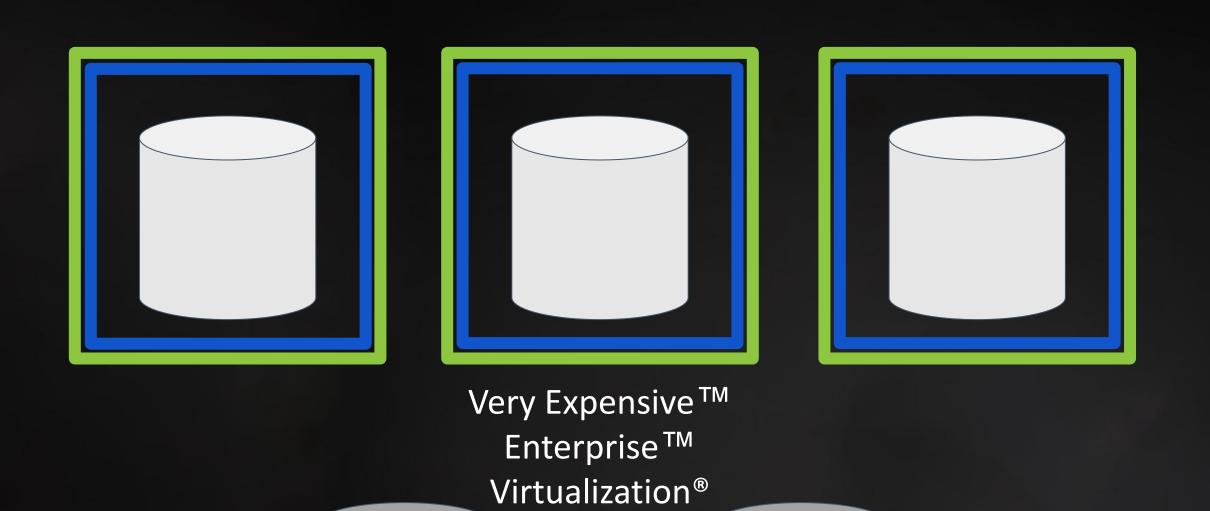
THE #1 PROGRAMMER EXCUSE FOR LEGITIMATELY SLACKING OFF:

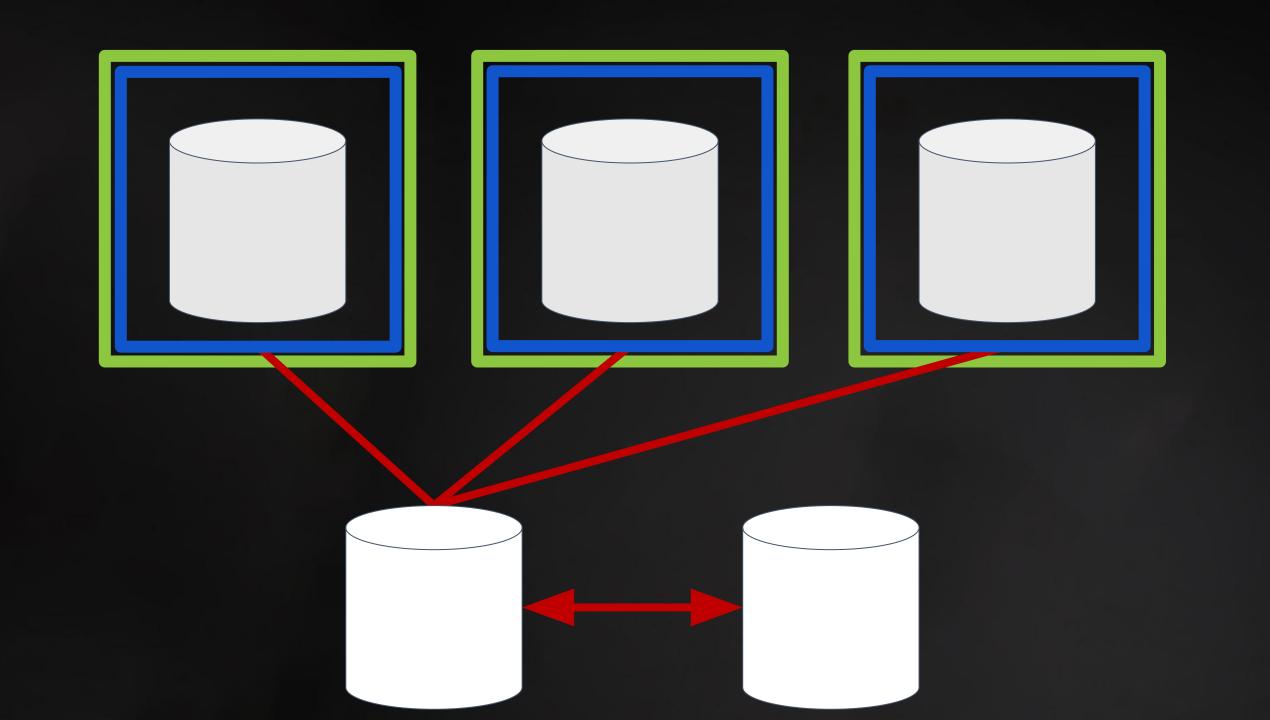
"MY CODE'S COMPILING."
MY DB IS INSTALLING.











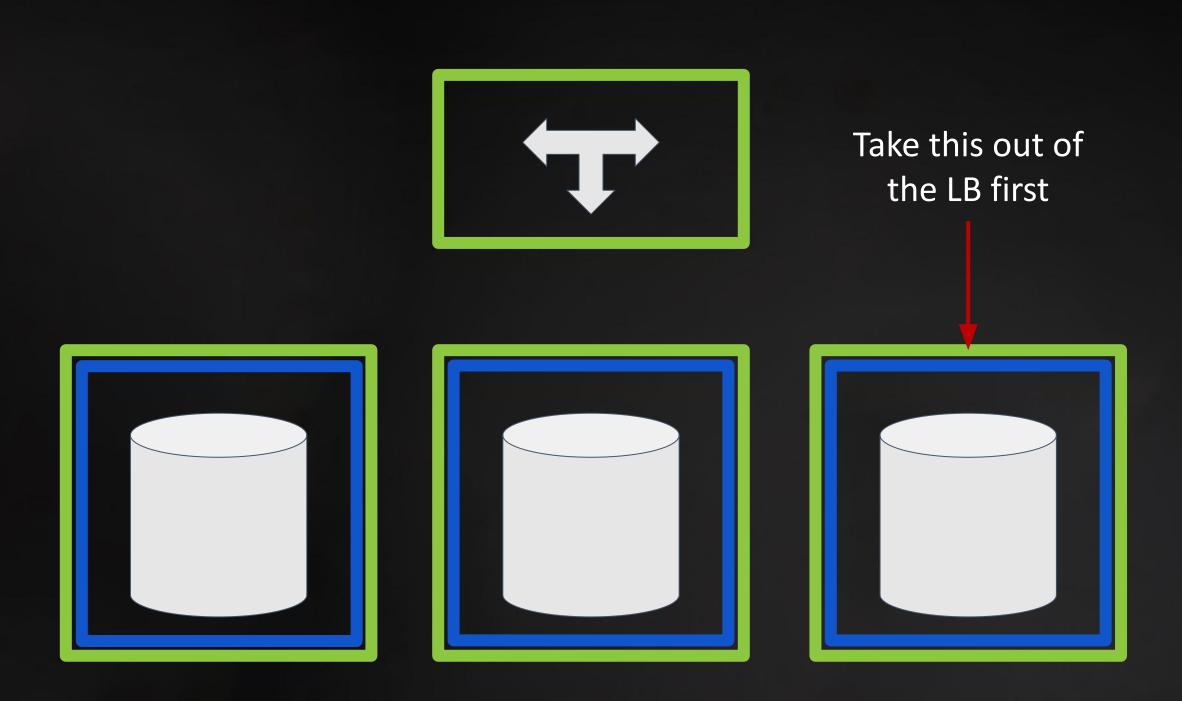


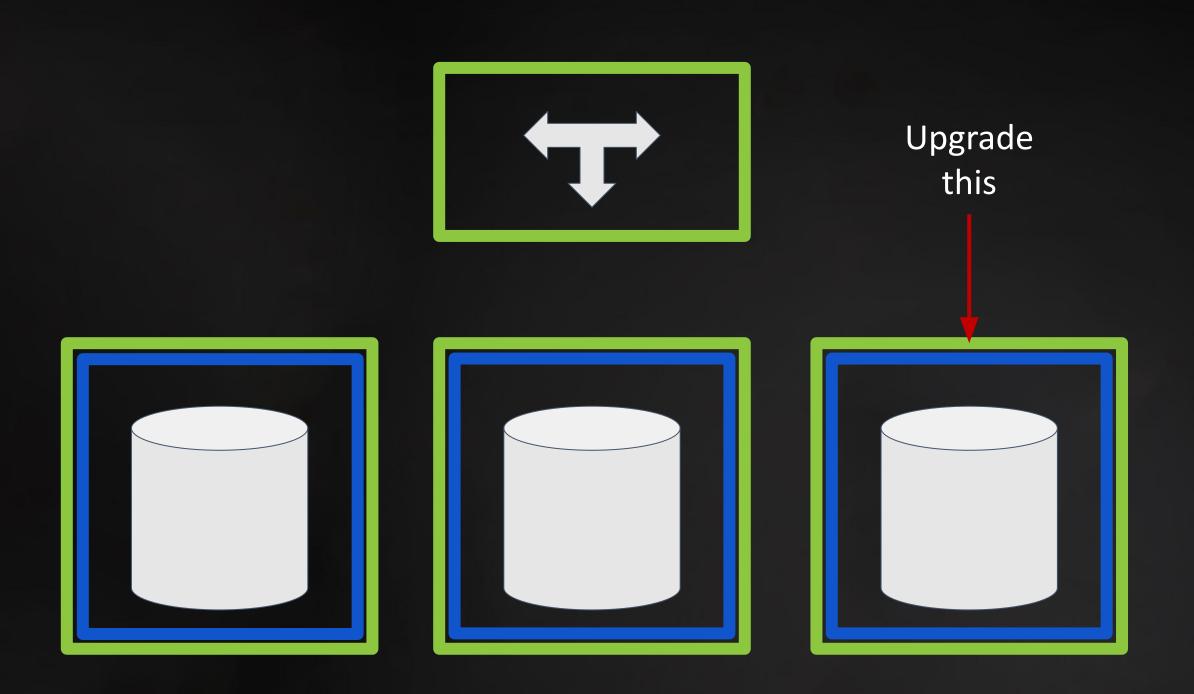
Very Expensive TM
Enterprise TM
Load Balancer®

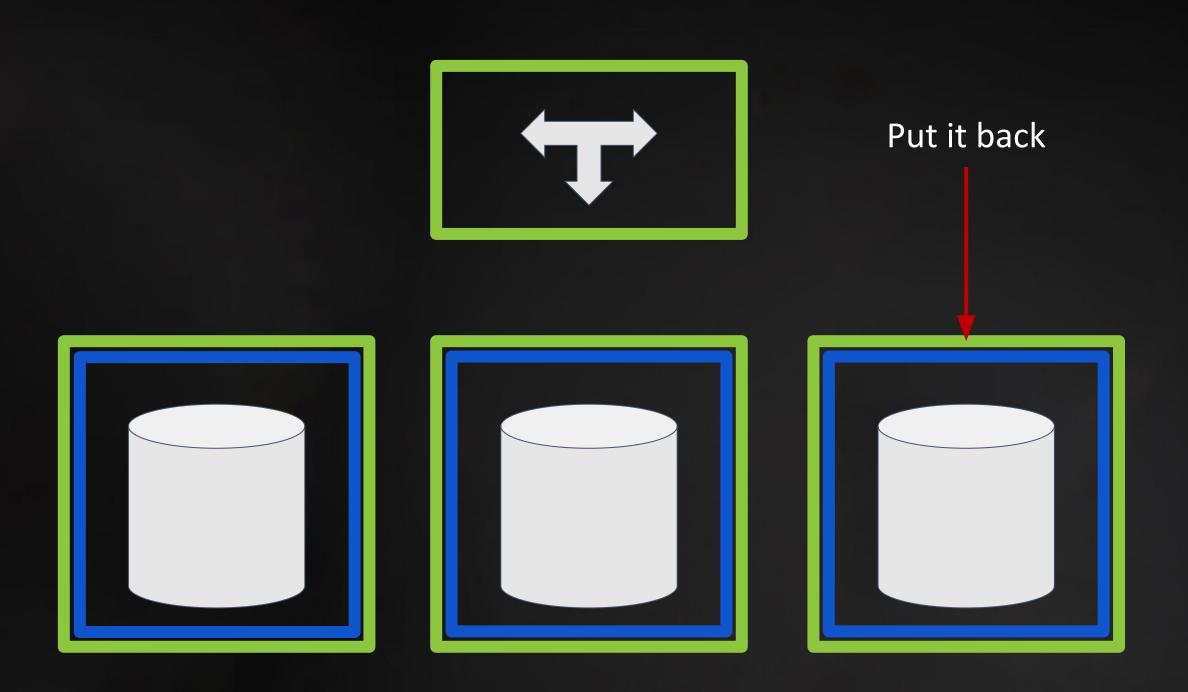














Then this







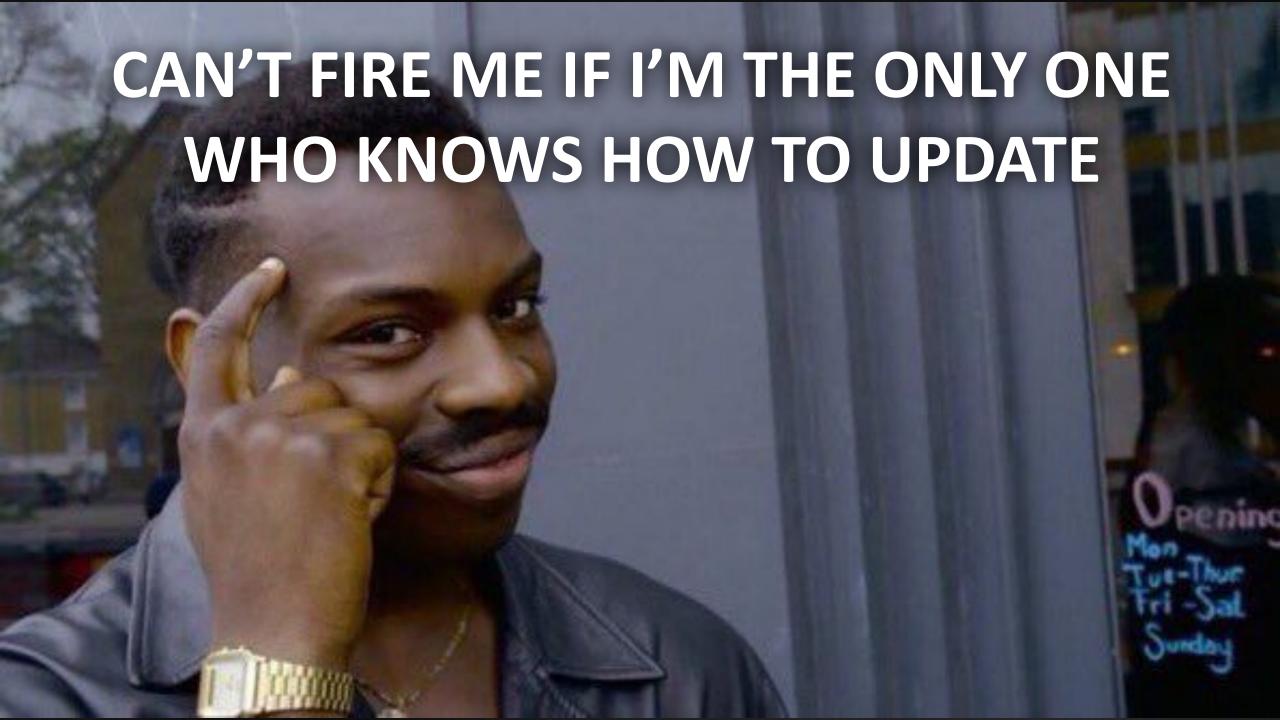


Then this



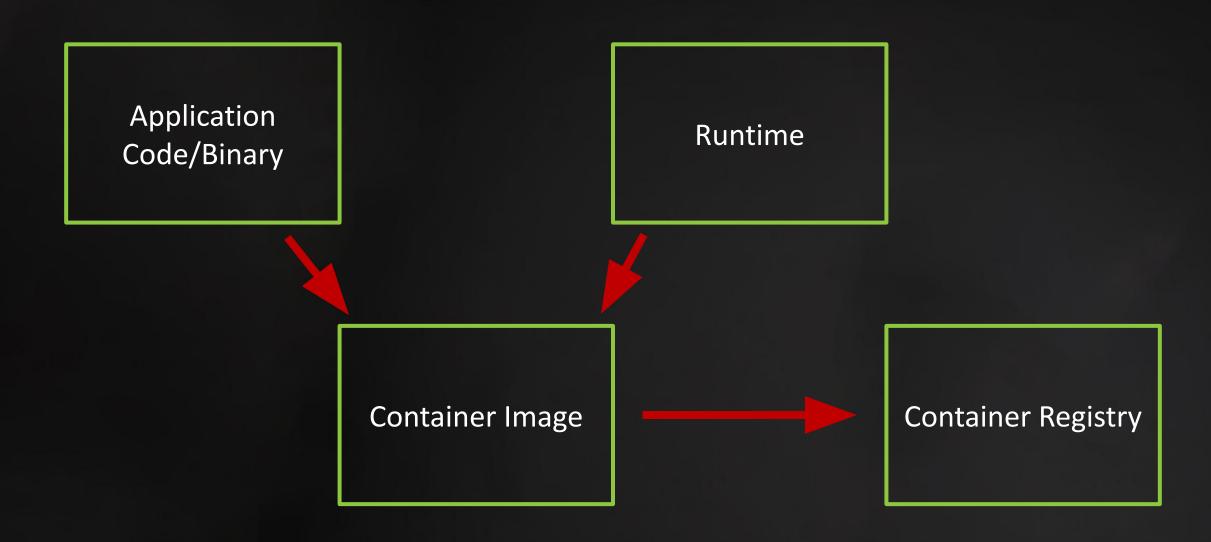






Containers

Container Image Build Process



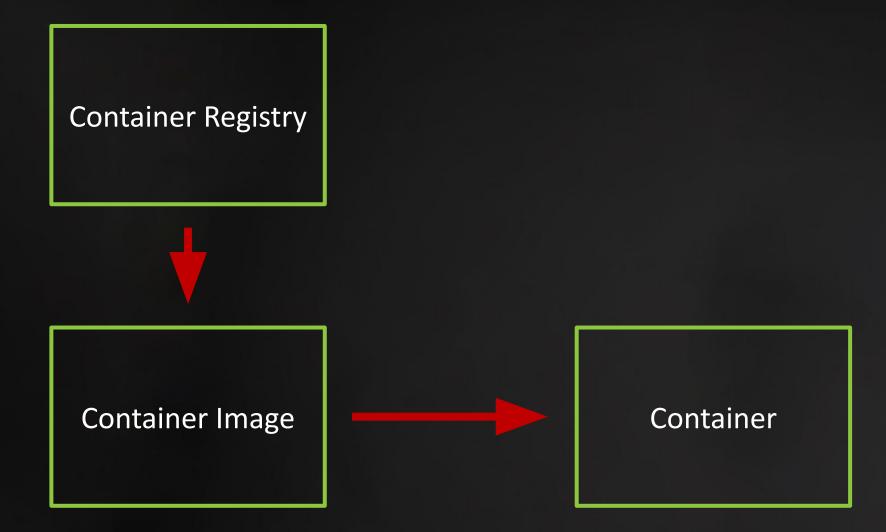
```
RUN apk upgrade --no-cache && \
    apk add --no-cache 'libssl1.1>=1.1.1k-r0' 'libcrypto1.1>=1.1.1k-r0'
COPY ./containerssh/containerssh-testauthconfigserver /
COPY ./containerssh/LICENSE.md /
COPY ./containerssh/NOTICE.md /
RUN chmod +x /containerssh-testauthconfigserver
ENTRYPOINT ["/containerssh-testauthconfigserver"]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

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

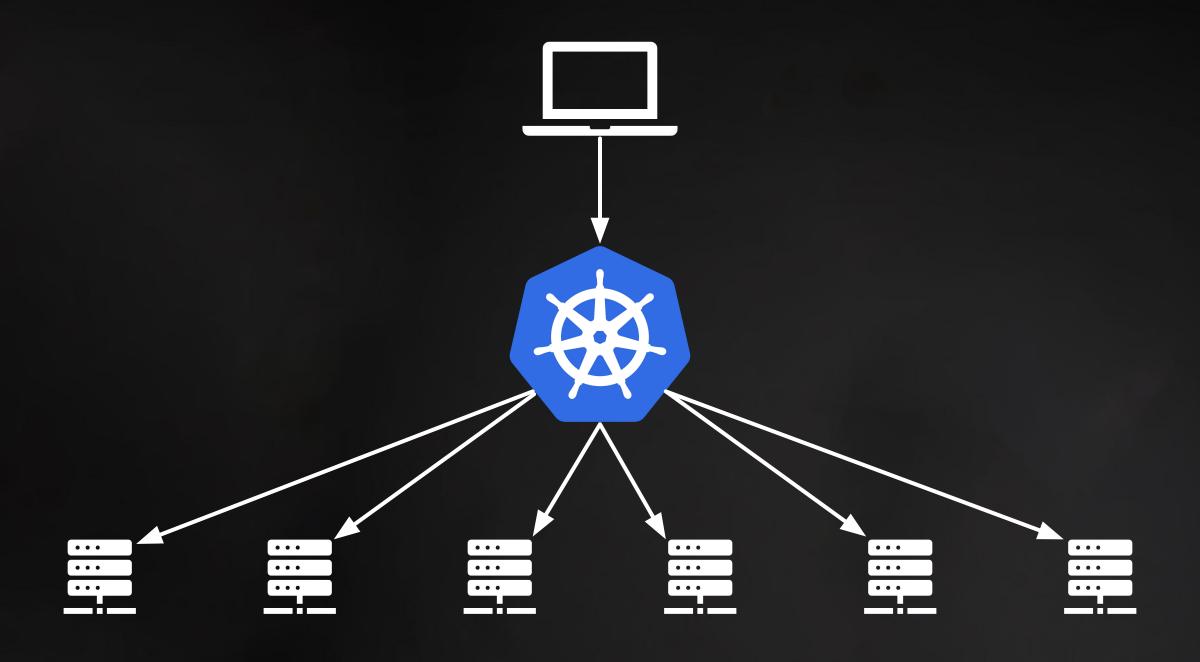
```
RUN apk upgrade --no-cache && \
    apk add --no-cache 'libssl1.1>=1.1.1k-r0' 'libcrypto1.1>=1.1.1k-r0'
COPY ./containerssh/containerssh-testauthconfigserver /
COPY ./containerssh/NOTICE.md /
RUN chmod +x /containerssh-testauthconfigserver
ENTRYPOINT ["/containerssh-testauthconfigserver"]
CMD []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

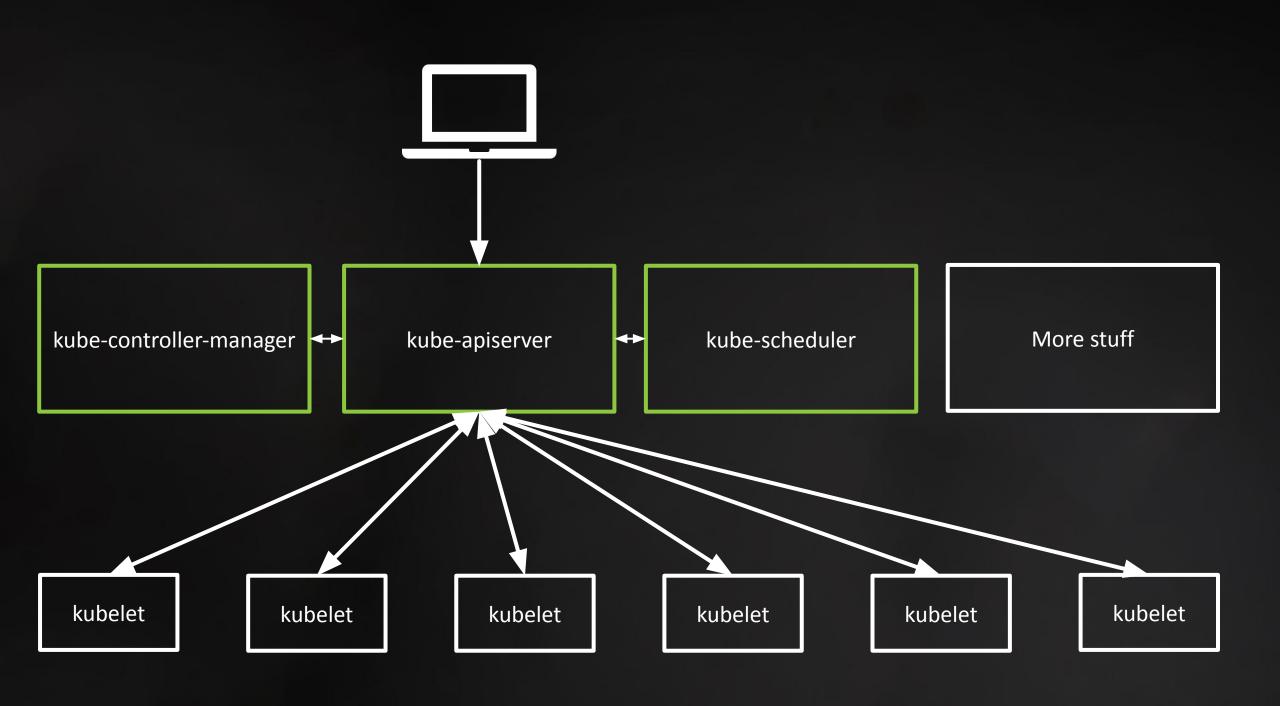
```
FROM alpine
RUN apk upgrade --no-cache && \
    apk add --no-cache 'libssl1.1>=1.1.1k-r0' 'libcrypto1.1>=1.1.1k-r0'
COPY ./containerssh/containerssh-testauthconfigserver /
COPY ./containerssh/NOTICE.md /
RUN chmod +x /containerssh-testauthconfigserver
ENTRYPOINT ["/containerssh-testauthconfigserver"]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

Running a Container









kubelet / worker kubelet / worker kubelet / worker kubelet / worker kubelet / worker

database

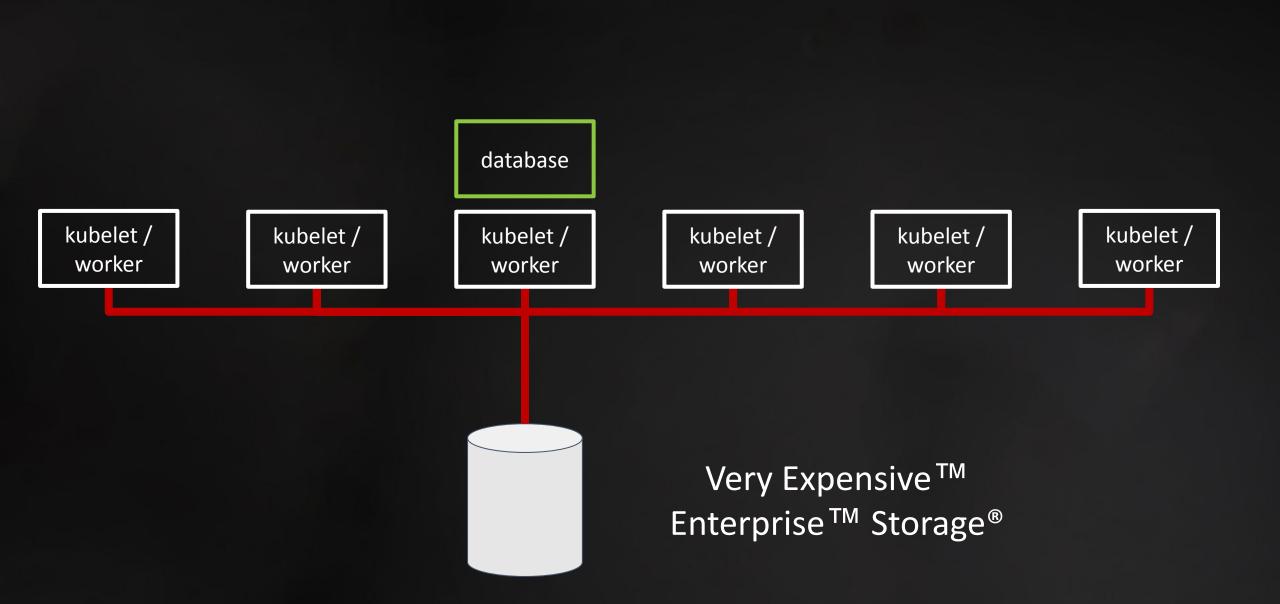
kubelet / worker kubelet / worker kubelet / worker kubelet / worker kubelet / worker

kubelet / worker kubelet / worker kubelet / worker kubelet / worker worker worker kubelet / worker worker worker

kubelet / worker kubelet / worker kubelet / worker kubelet / worker

database

kubelet / worker kubelet / worker kubelet / worker kubelet / worker kubelet / worker



database kubelet / kubelet / kubelet / kubelet / kubelet / worker worker worker worker worker Very Expensive TM Enterprise TM Storage®

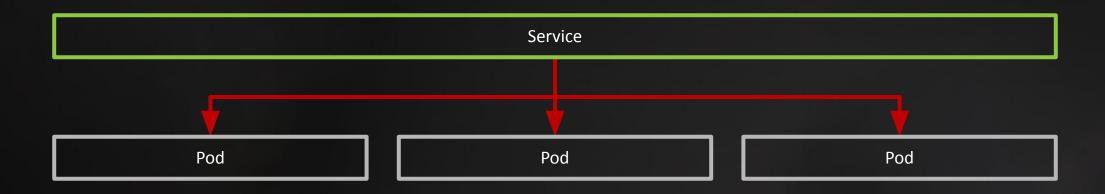


```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: db-pvc-001
spec:
  resources:
    requests:
      storage: 80Gi
accessModes:
      - ReadWriteOnce
```

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: db-pvc-001
spec:
  resources:
    requests:
    storage: 80Gi
accessModes:
  - ReadWriteOnce
```

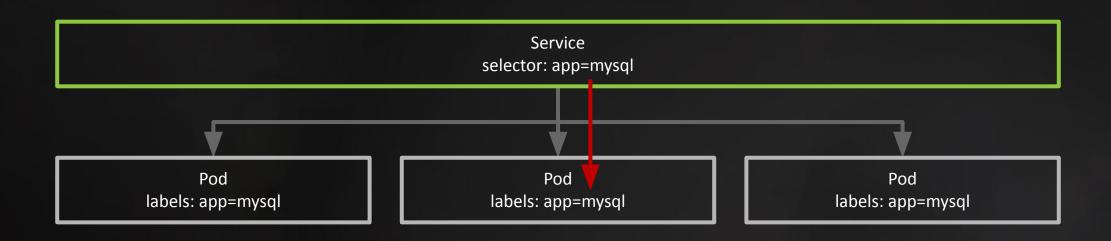
```
apiVersion: v1
kind: PersistentVolume
metadata:
 name: db-pv-001
spec:
 capacity:
   storage: 80Gi
 volumeMode: Filesystem
 accessModes:
   - ReadWriteOnce
 nfs:
   path: /tmp
   server: 172.17.0.2
mountOptions:
   - hard
   - nfsvers=4.1
```

```
apiVersion: v1
kind: Pod
metadata:
 name: mysql-001
spec:
 containers:
   - image: mysql:5.6
     name: mysql
     env:
       - name: MYSQL ROOT PASSWORD
         value: changeme
     ports:
       - containerPort: 3306
         name: mysql
     volumeMounts:
       - name: mysql-persistent-storage
         mountPath: /var/lib/mysql
 volumes:
   - name: mysql-persistent-storage
     persistentVolumeClaim:
       claimName: db-pvc-001
```



```
apiVersion: v1
kind: Pod
metadata:
  name: mysql-001
labels:
  app: mysql
spec:
  //...
```

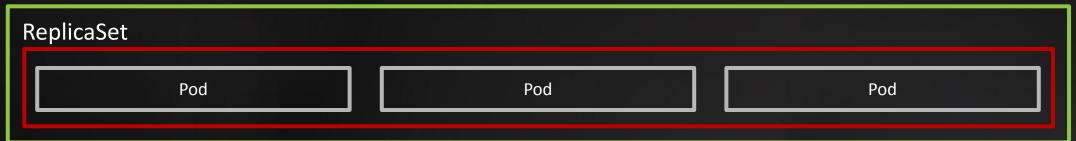
```
apiVersion: v1
kind: Service
metadata:
  name: mysql
spec:
  ports:
    - name: mysql
       port: 3306
      targetPort: 3306
selector:
    app: mysql
```

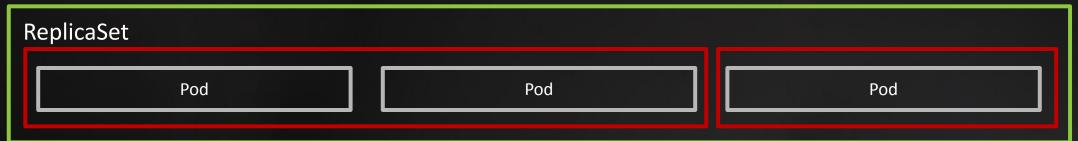


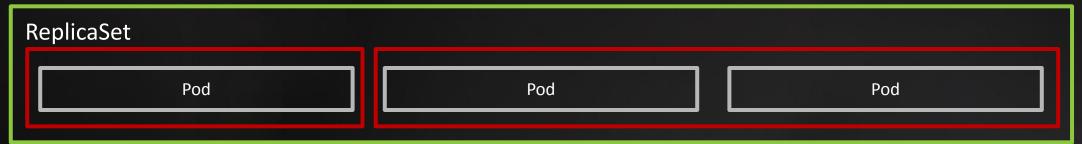
Pod Pod Pod

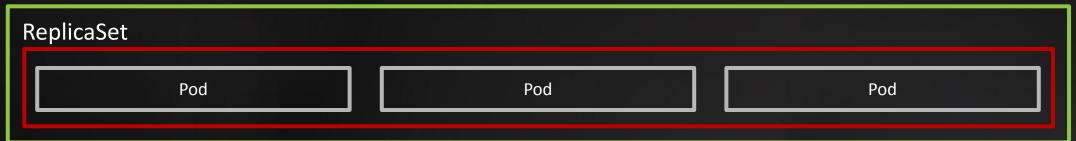
Pod Pod Pod

ReplicaSet Pod Pod Pod Pod



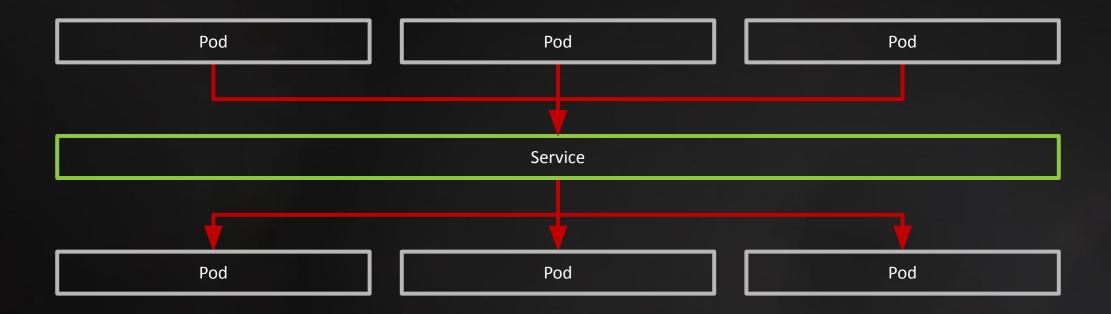






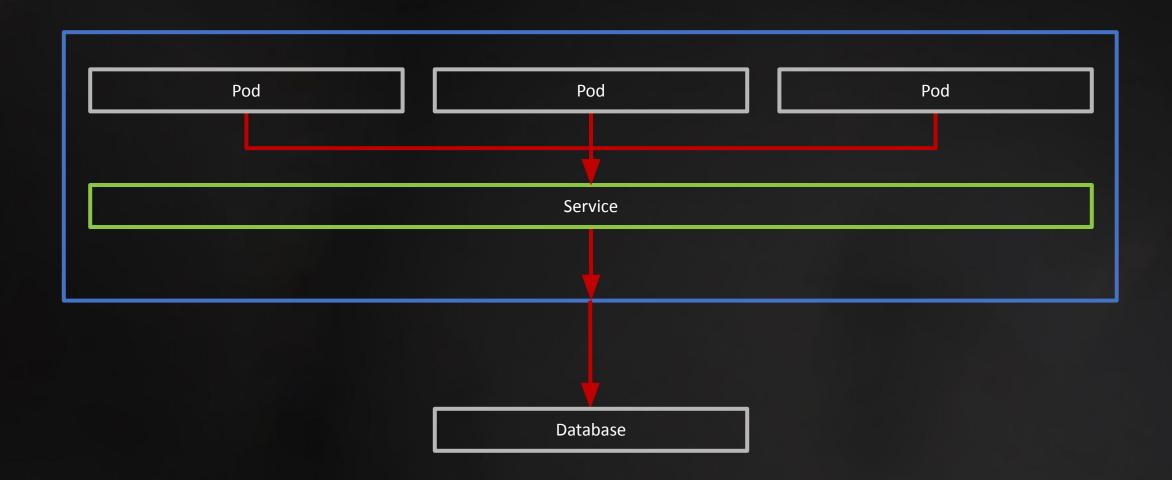
StatefulSet

Pod 1 Pod 2 Pod 3



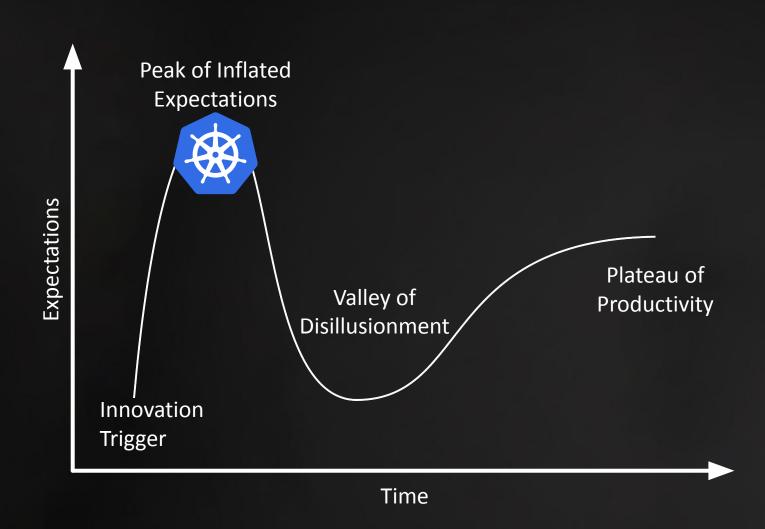
```
apiVersion: v1
kind: Service
metadata:
name: mysql
spec:
type: LoadBalancer
 ports:
   - name: mysql
     port: 3306
     targetPort: 3306
   app: mysql
```

```
apiVersion: v1
kind: Service
metadata:
name: mysql
spec:
type: NodePort
 ports:
   - name: mysql
     port: 3306
     targetPort: 3306
   app: mysql
```



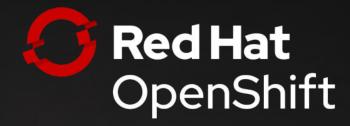
```
apiVersion: v1
kind: Service
metadata:
  name: mysql
spec:
  type: ExternalName
  externalName: debugged.it
  ports:
    - name: mysql
      port: 3306
      targetPort: 3306
```

The Gartner Hype Cycle





















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



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

Thank you for listening!