CNUTCon 2017 全球运维技术大会

# 分布式数据库系统 TiDB 在 Kubernetes 平台的自动化运维实践

邓栓

PingCAP SRE 工程师

主办方 Geekbang 极客邦科技 InfoQ

Geekbang.
极客邦科技

# 极客搜索

全站干货，一键触达，只为技术

s.geekbang.org

扫描二维码立即体验

有没有一种搜索方式，能整合 InfoQ 中文站、极客邦科技旗下12大微信公众号矩阵的全部资源?
极客搜索，这款针对极客邦科技全站内容资源的轻量级搜索引擎，做到了!

**扫描上方二维码，极客搜索!**

EGO EXTRA GEEKS' ORGANIZATION NETWORKS

# 这里只有
# 技术领导者

## EGO会员第二季招募季正式开启


报名时间：9月1日-9月15日

扫描添加E小欧，

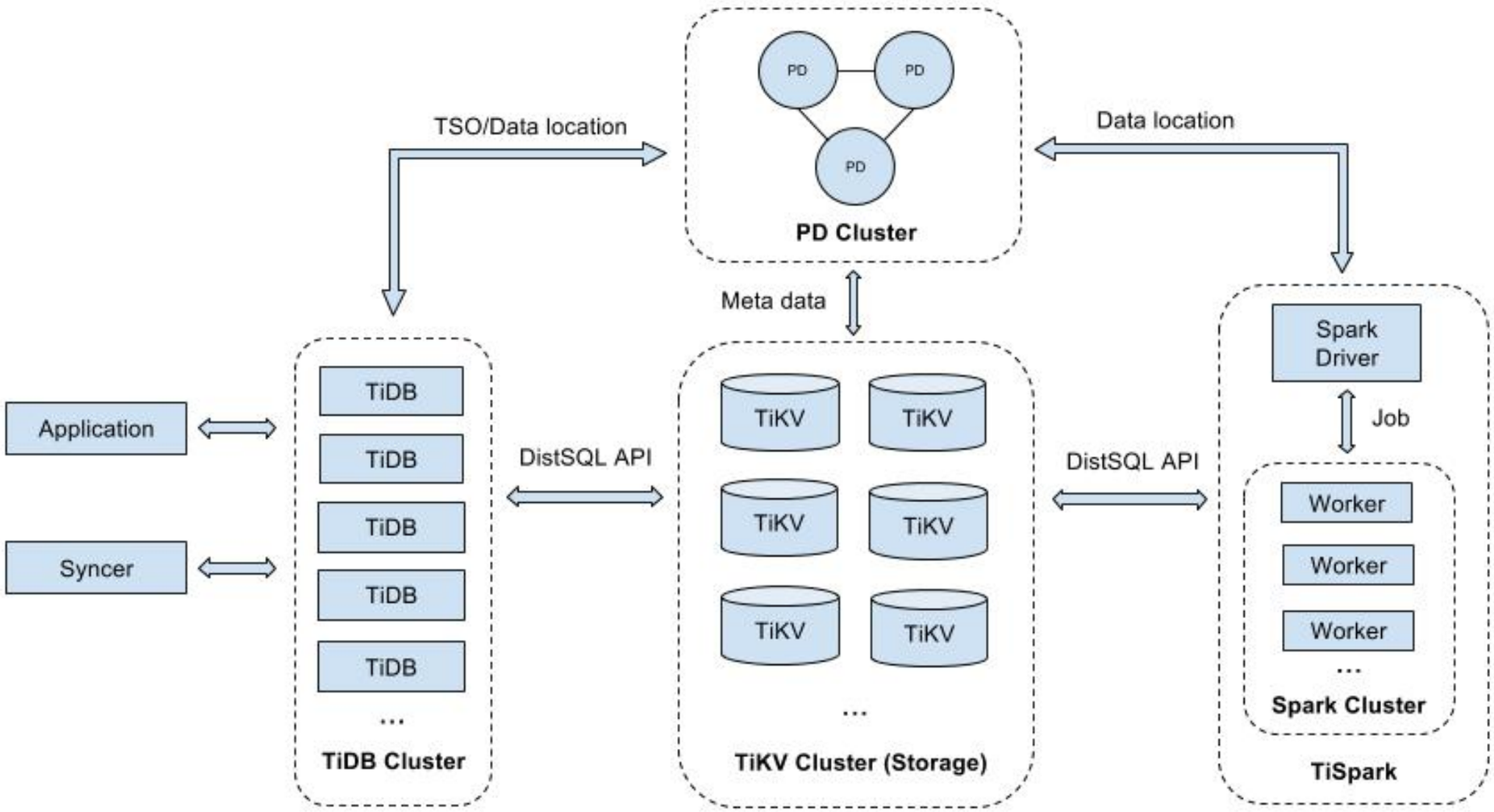
邀您进入EGO会员预报名群

立即报名

E小欧

# TABLE OF **CONTENTS**

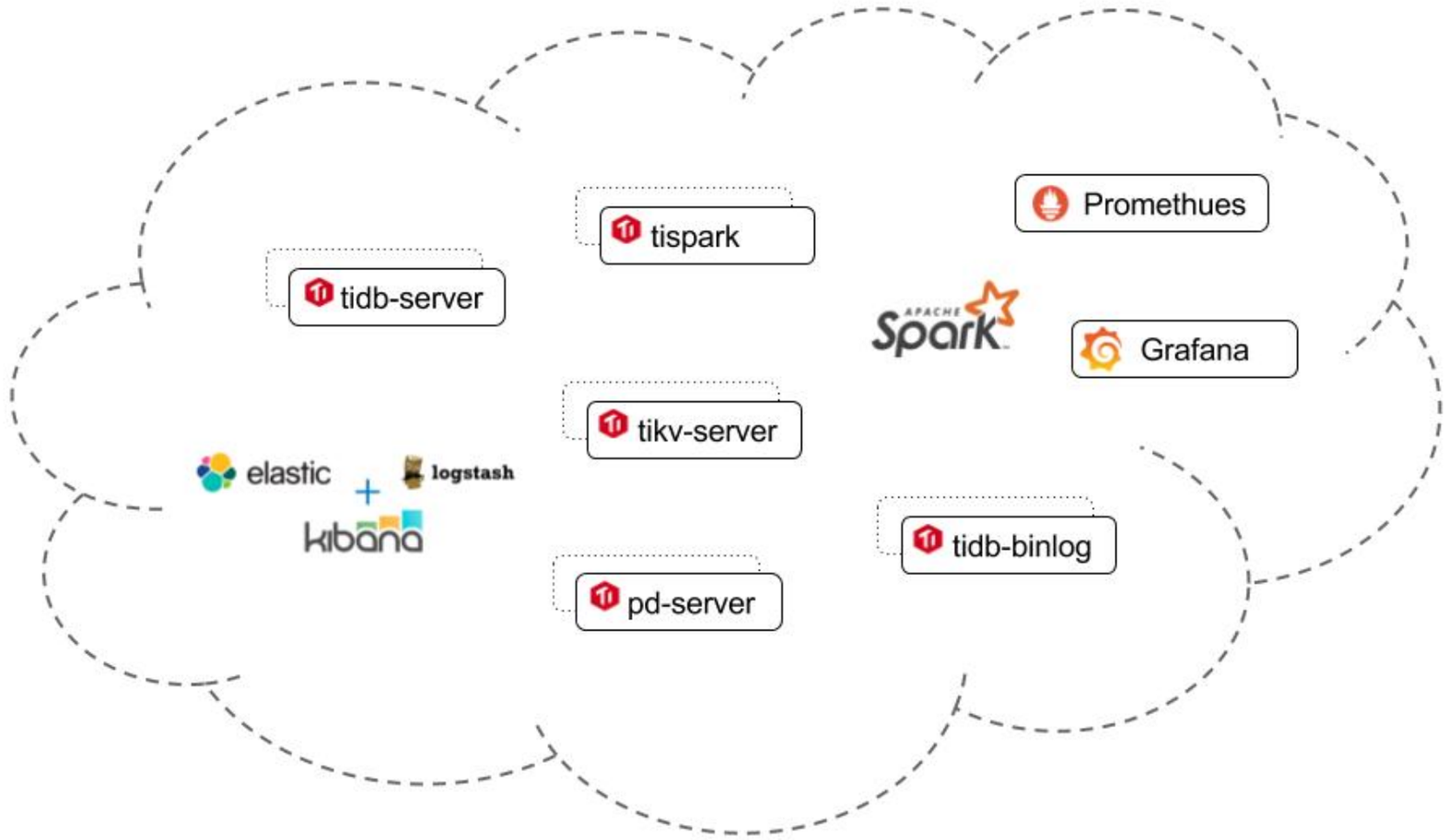
# 分布式 NewSQL 数据库 TiDB



# 分布式系统特点

- 伸缩，冗余，容灾
- 组件多，结构复杂

# 分布式系统部署和运维的挑战

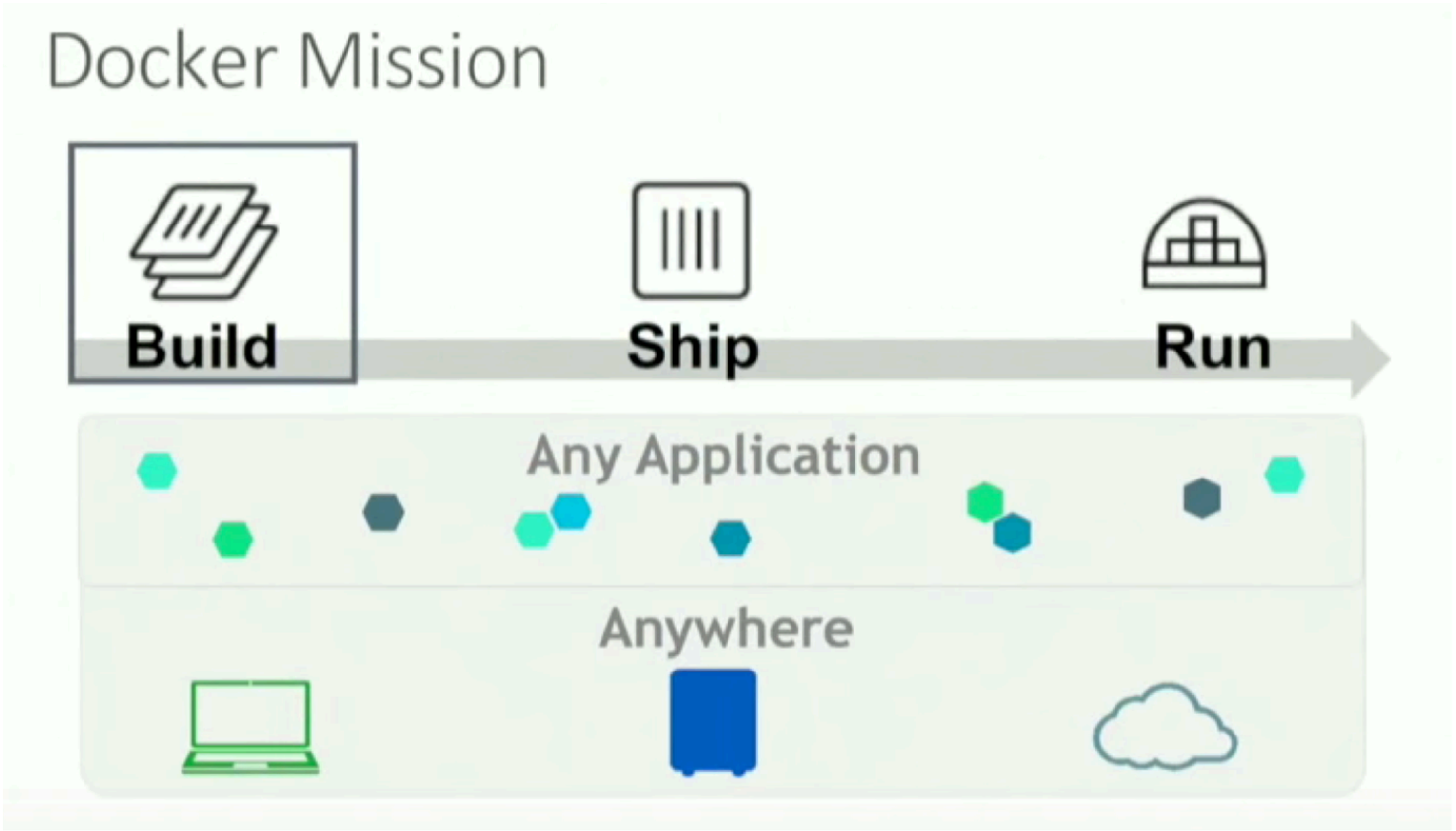


# 传统部署运维工具

- Puppet/Ansible/Chef/SaltStack
- 部署方式复杂，脚本配合 DSL 描述文件
- 状态管理功能简单，缺乏全局调度管理

# 服务发布和部署容器化

- 开发环境与运行环境一致
- 服务发布标准化流程化
- 隔离性



# 容器编排系统

- Kubernetes
- Docker Swarm
- Mesos & Marathon

# 我们的选择: Kubernetes

- 基于 Google Borg 的开源容器编排工具
- 开发社区极其活跃，容器编排系统中最流行
- 方便与各种云平台整合，云上操作系统

# TABLE OF **CONTENTS**

# kubernetes 数据存储

- Docker volume: 显示指定路径和Dockerfile指定volume
- k8s 持久化存储(PV): EBS, NFS, Ceph, Glusterfs…
- k8s 本地存储: emptyDir, hostPath

# Kubernetes 有状态服务的管理
# StatefulSet

- 需要 PV(PersistentVolume)，网络文件系统性能问题
- 状态维护过于简单，只保证顺序
- 分布式系统不同组件之间协调关系没法保障，缺乏特定系统运维知识

# TABLE OF **CONTENTS**

# k8s 工作原理

* kubectl apply -f example.yaml

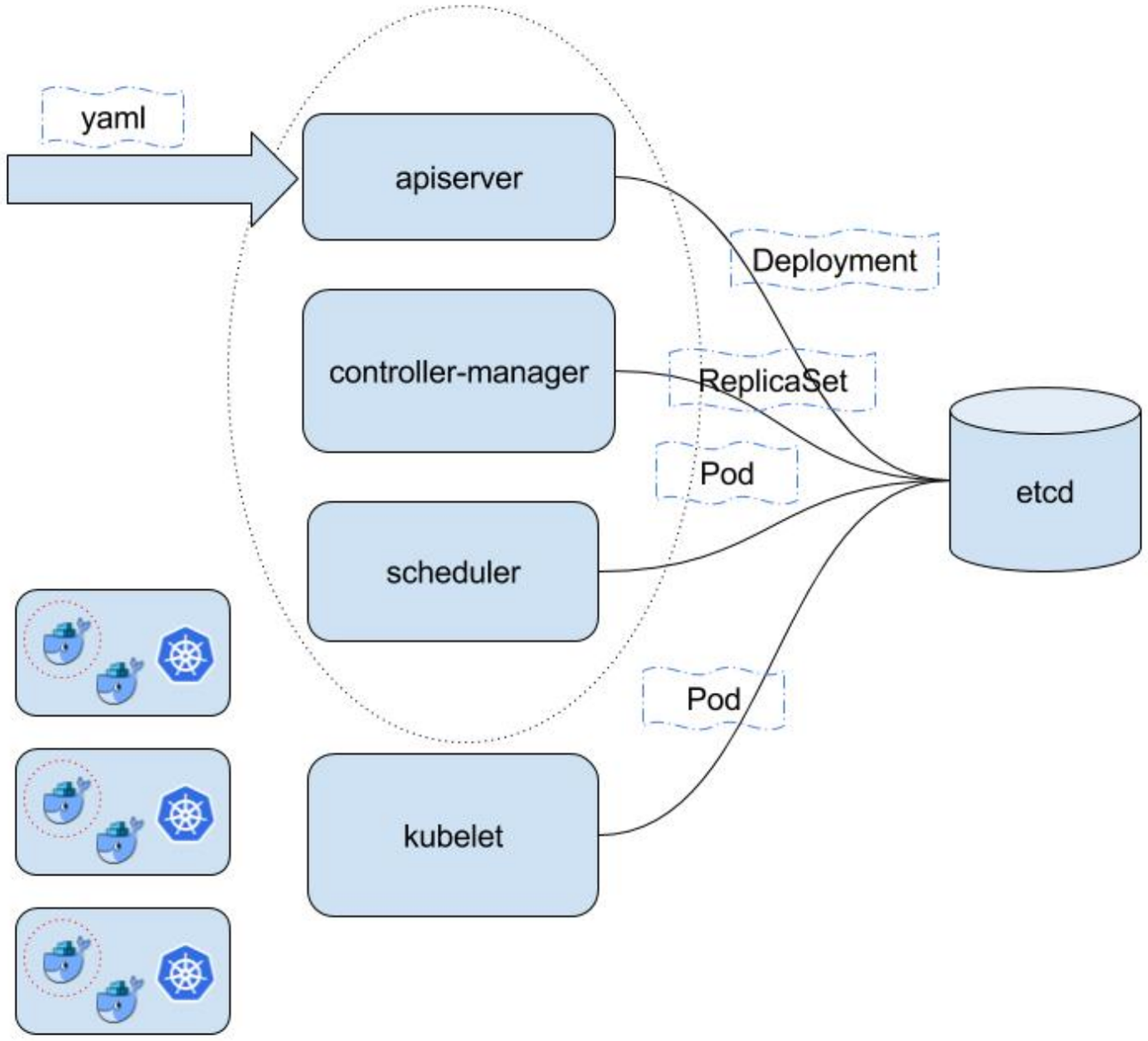
* kubectl get deployments

* kubectl get replicasets

* kubectl get pods

```yaml
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: nginx
spec:
  affinity:
    podAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
      - labelSelector:
          matchExpressions:
          - key: security
            operator: In
            values:
            - S1
        topologyKey: failure-domain.beta.kubernetes.io/zone
  replicas: 3
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: with-pod-affinity
      image: nginx:1.7.9
      resources:
        limits:
          cpu: 400m
          memory: 512M
        requests:
          cpu: 250m
          memory: 200M
      ports:
      - containerPort: 80
```

# k8s 工作原理


yaml
apiserver
controller-manager
scheduler
kubelet
Deployment
ReplicaSet
Pod
Pod
etcd

# Kubernetes Operator 模式

- 自定义资源类型(TPR/CRD)
- 扩展 k8s controller，引入特定系统运维知识
- 运维自动化：upgrade, scale, failover, backup
- 不涉及数据持久化

# TABLE OF **CONTENTS**

# 有状态服务的运维

- 创建集群: PD join
- 增加节点
- 升级服务: PD -> TiKV -> TiDB
- 下线节点: 加节点 -> 从 PD 下线节点 -> 删除节点

# Operator 模式实践: TiDB-Operator

- 保证 TiDB 不同组件启动和升级顺序(PD->TiKV->TiDB)
- 扩展 k8s 内置的调度器
- 按正确方式扩容和缩容
- 按组件类型处理节点故障
- 使用本地存储提高 IO 性能

# TABLE OF **CONTENTS**

# TiDB-Operator 架构

* kubectl apply -f demo-cluster.yaml

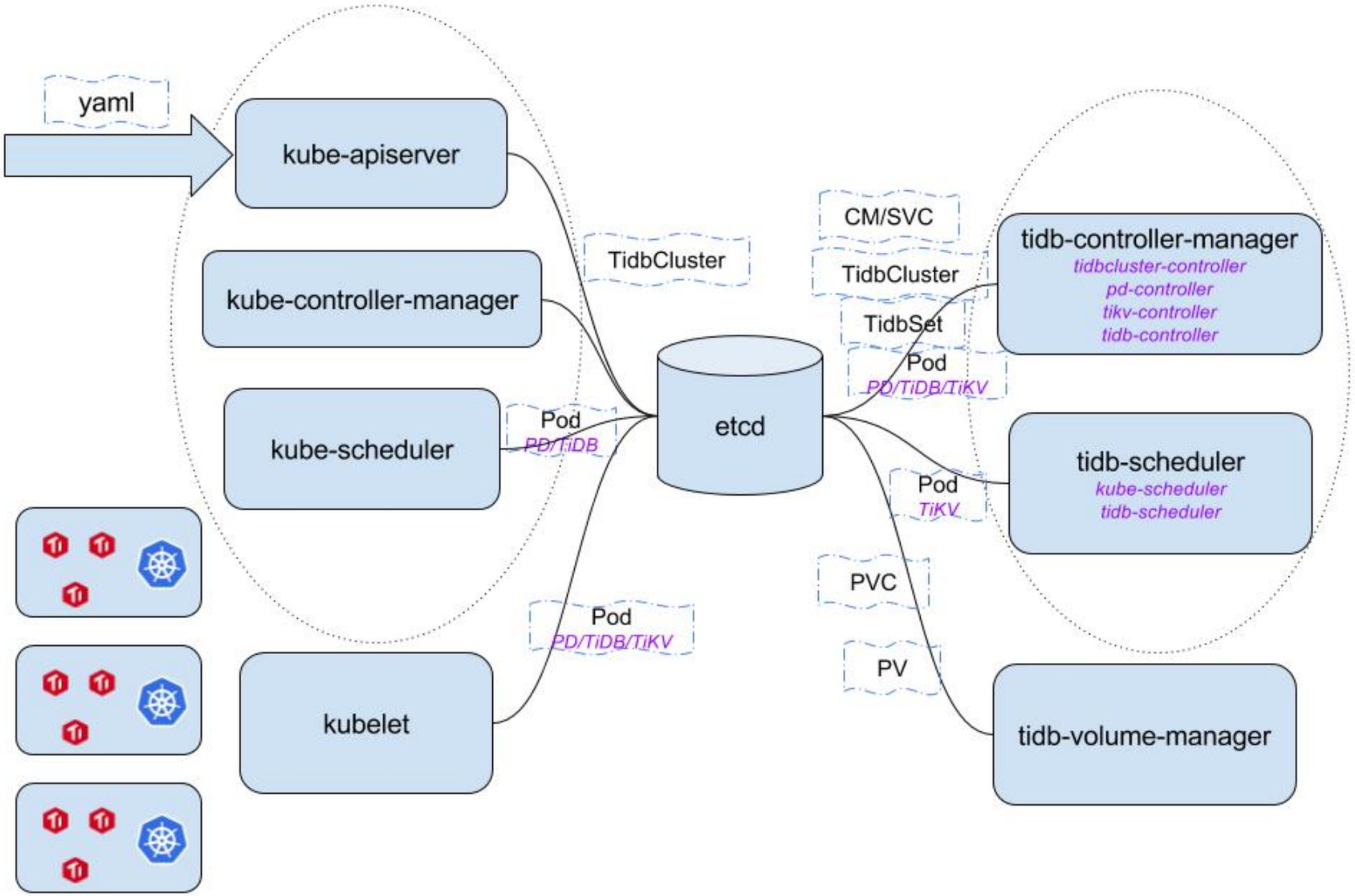
* kubectl get tidbclusters

* kubectl get tidbsets

* kubectl get pods, services

```yaml
apiVersion: pingcap.com/v1
kind: TidbCluster
metadata:
  name: demo-cluster
spec:
  pd:
    size: 1
    image: pingcap/pd:latest
  tidb:
    size: 2
    image: pingcap/tidb:latest
  tikv:
    size: 3
    image: pingcap/tikv:latest
  service: NodePort
```

# TiDB-Operator 架构



# TABLE OF **CONTENTS**

# TiDB-Operator 实现

- tidb-controller-manager
- tidb-scheduler
- tidb-volume-manager

# tidb-controller-manager

- 扩展 k8s 内置资源 CRD: TidbCluster, TidbSet
- 内嵌 TiDB 运维知识的 Controller:
  * tidbcluster-controller
  * pd-controller
  * tikv-controller
  * tidb-controller
  * gc-controller
- 数据层面和实例层面两层保障节点失效时服务的可用性

# tidb-scheduler

- 利用 k8s 内置 scheduler 实现基本调度(cpu/memory/affinity)
- 扩展 k8s scheduler 实现基于 PV 的调度
- 结合 PD 数据调度规则优化 TiKV 实例调度(两层调度最大限度保证数据可用性)

# tidb-volume-manager

- 基于 external-storage 实现 PV 的管理
- 基于 hostPath 实现 Local PV (StorageClass: pingcap-volume-provisioner)

# CNUTCon TiDB 交流讨论

# THANKS!

智 能 时 代 的 新 运 维

CNUTCon 2017