# TiDB on Kubernetes

Huang Dongxu

CTO, PingCAP

# About Me

- Huang Dongxu (黄东旭)
- CTO, Co-founder of PingCAP
- Infrastructure Engineer / Open-source advocator
- Co-author of Codis / TiDB / TiKV
- MSRA => Netease => WandouLabs => PingCAP
- h@pingcap.com

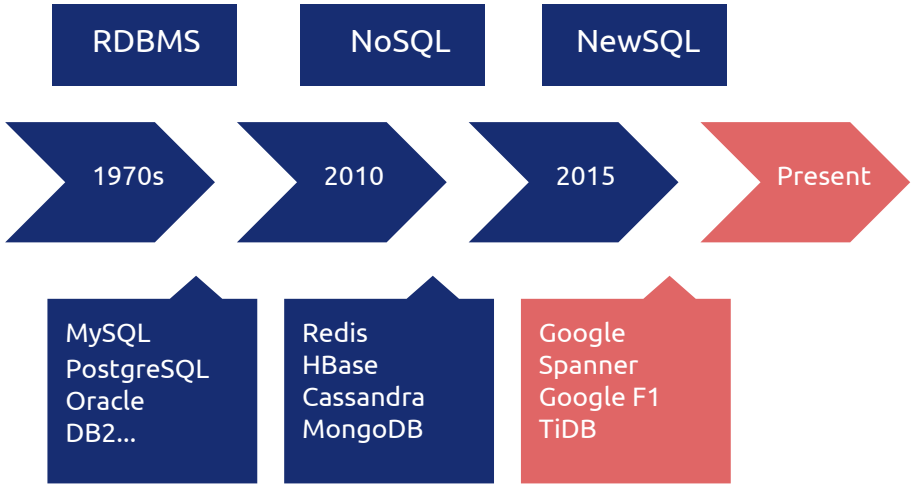
# Agenda

- TiDB introduction
  - TiDB architecture
  - TiDB ecosystem
- Why combine TiDB & Kubernetes
  - Cloud vendor agnostic
  - Automation
- How we make it possible
  - TiDB Operator architecture & features
  - How we manage state
  - How we schedule stateful app

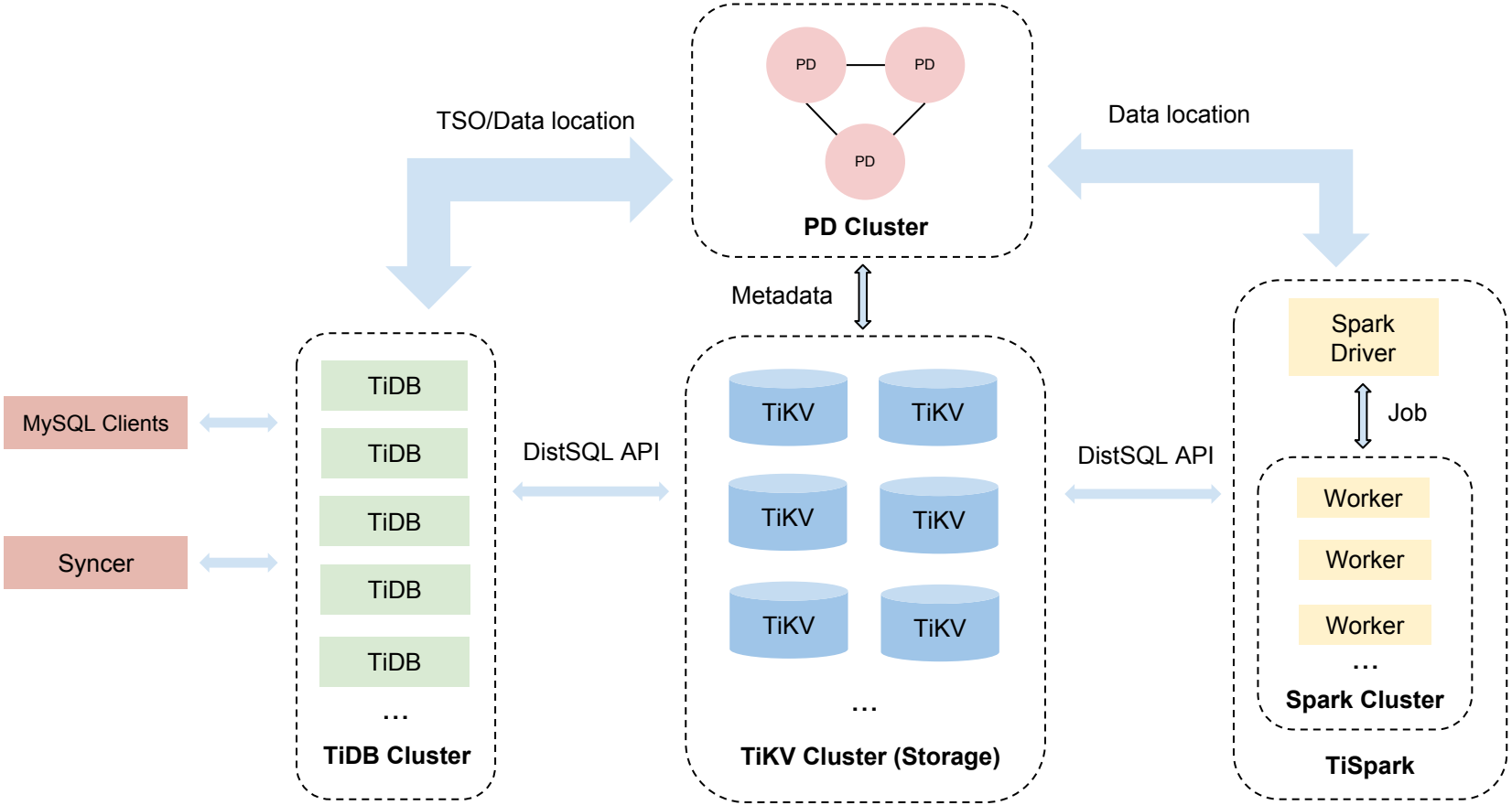
# Part I - Intro to TiDB

# Why we want to build a NewSQL Database

- From the beginning
- What's wrong with the existing DBs?
  - RDBMS
  - NoSQL & Middleware
- NewSQL:  F1 & Spanner

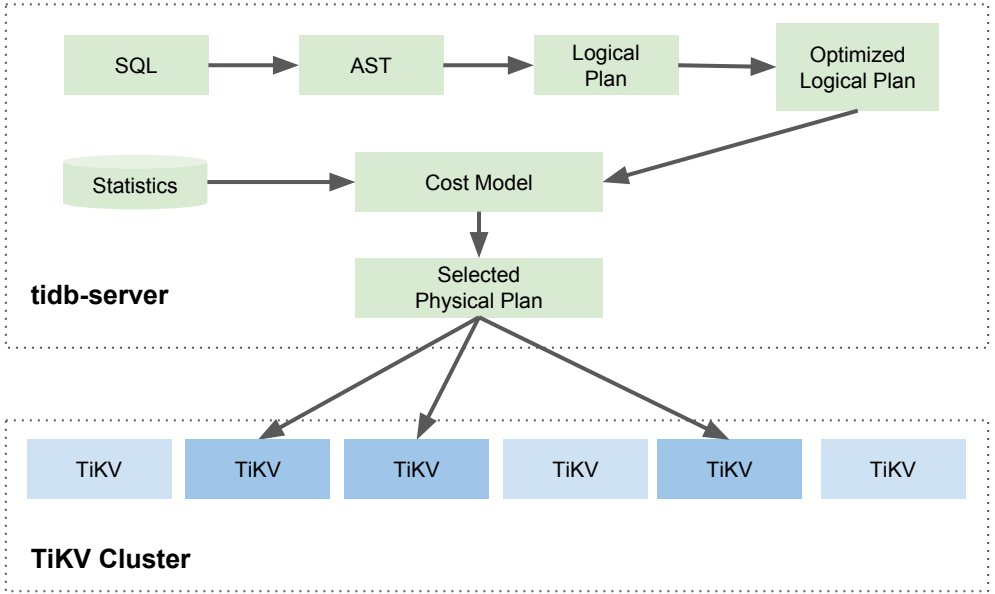


# TiDB architecture



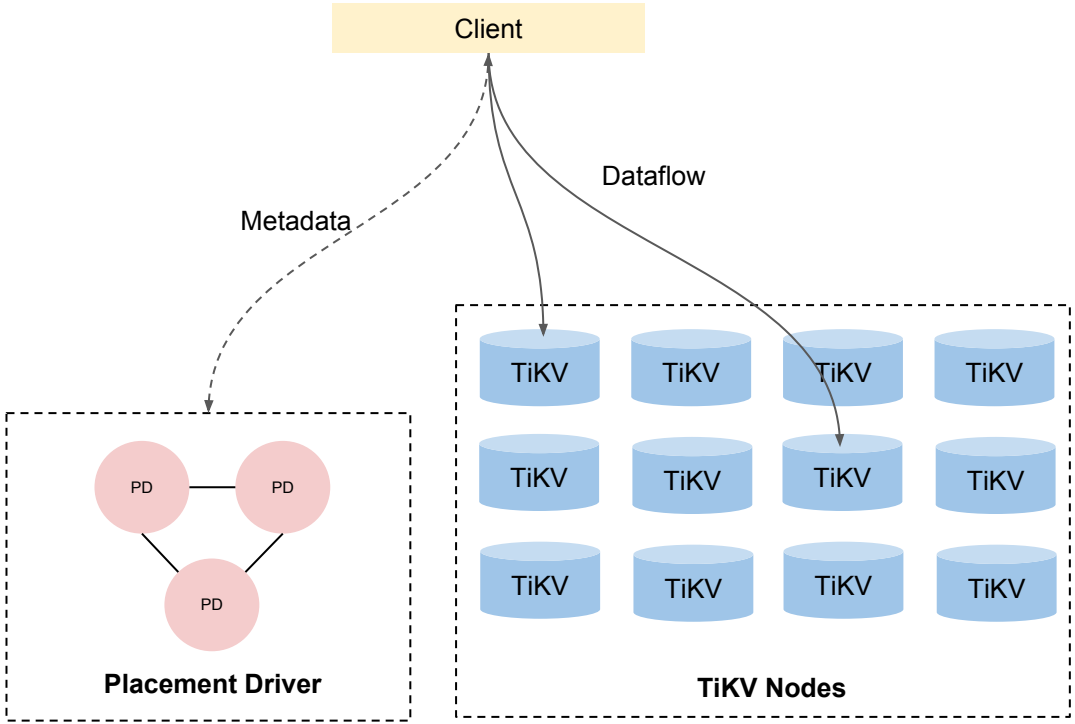
# TiDB: Computing

- Stateless SQL layer
  - Client can connect to any existing tidb-server instance
  - TiDB *__will not__* re-shuffle the data across different tidb-servers
- Full-featured SQL Layer
  - Speak MySQL wire protocol
    - Why not reusing MySQL?
  - Homemade parser  & lexer
  - RBO & CBO
  - Secondary index support
  - DML & DDL

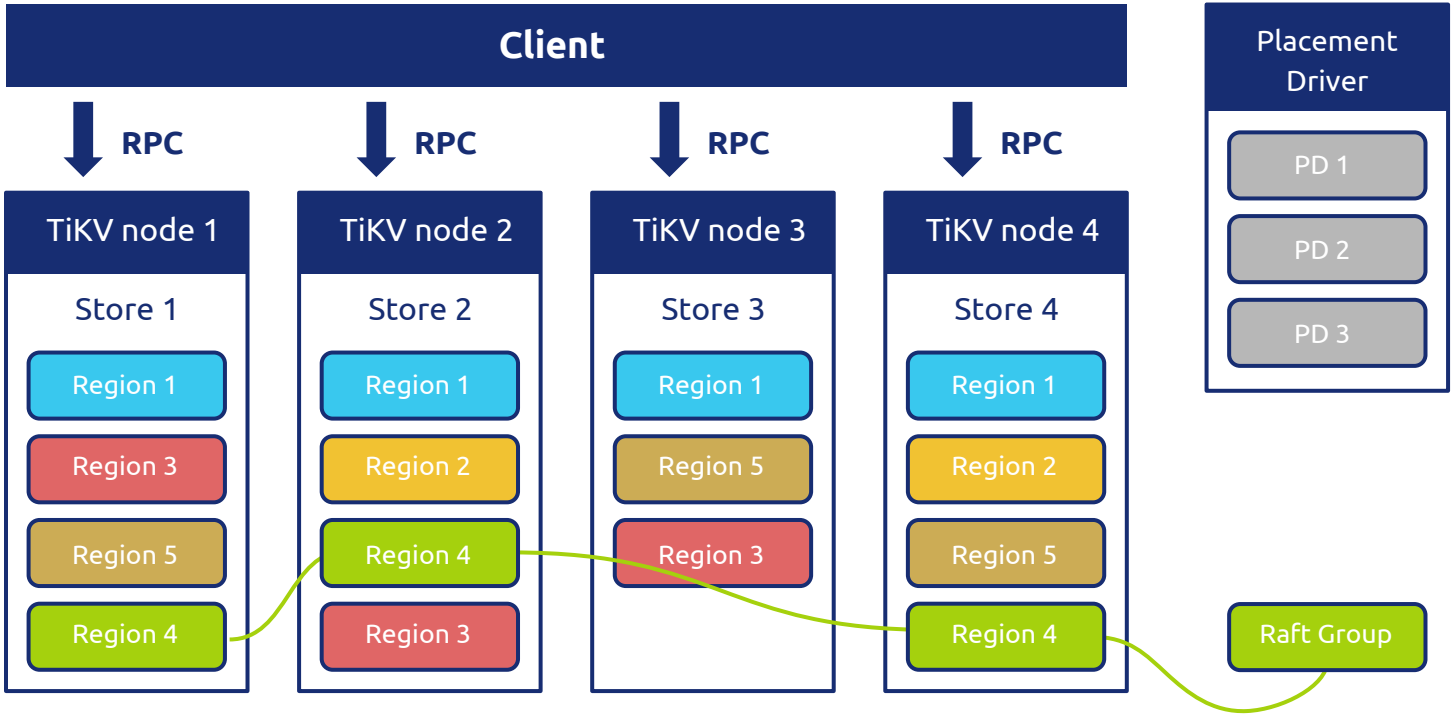


# TiKV: The Storage

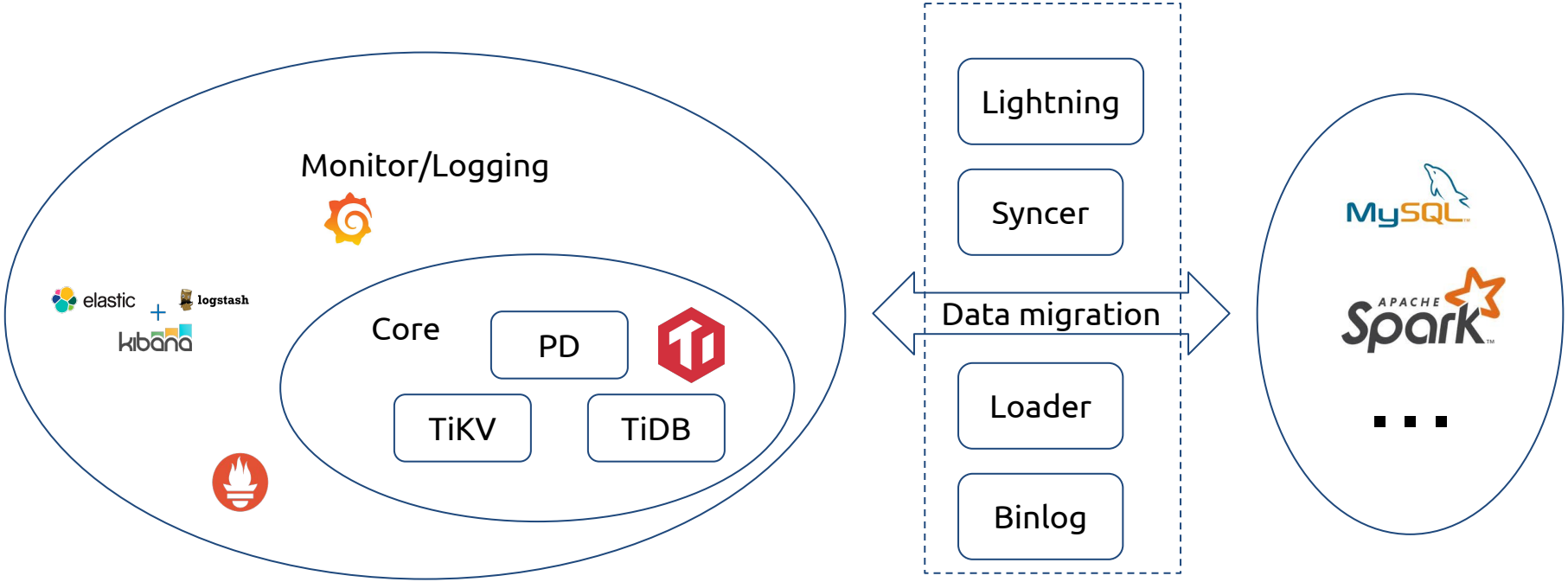
- The **storage layer** for TiDB
- Distributed **Key-Value** store
  - **Support ACID Transactions**
  - Replicate logs by **Raft**
  - **Range** partitioning
    - Split / merge **dynamically**
  - Support coprocessor for **SQL operators pushdown**



# TiKV: The Storage



# TiDB ecosystem



# Part II - Why TiDB on Kubernetes

# Cloud-Native applications

- Microservice architecure
- Easy deployment on any cloud
- Elastic scaling
- Highly available
- Automatic operation

Google Cloud

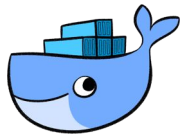

Microsoft Azure

# Kubernetes: standard platform

De-facto Container Orchestration System (Google Sponsored)

Distributed, cloud provider agnostic OS

- CPU, Memory, Storage and other Devices across all nodes
- Container <==> Process
- Docker image <==> Executable artifacts
- Deployment, StatefulSet  <==> Systemd/Supervisor …
- Helm / Charts <==> apt yum / deb rpm

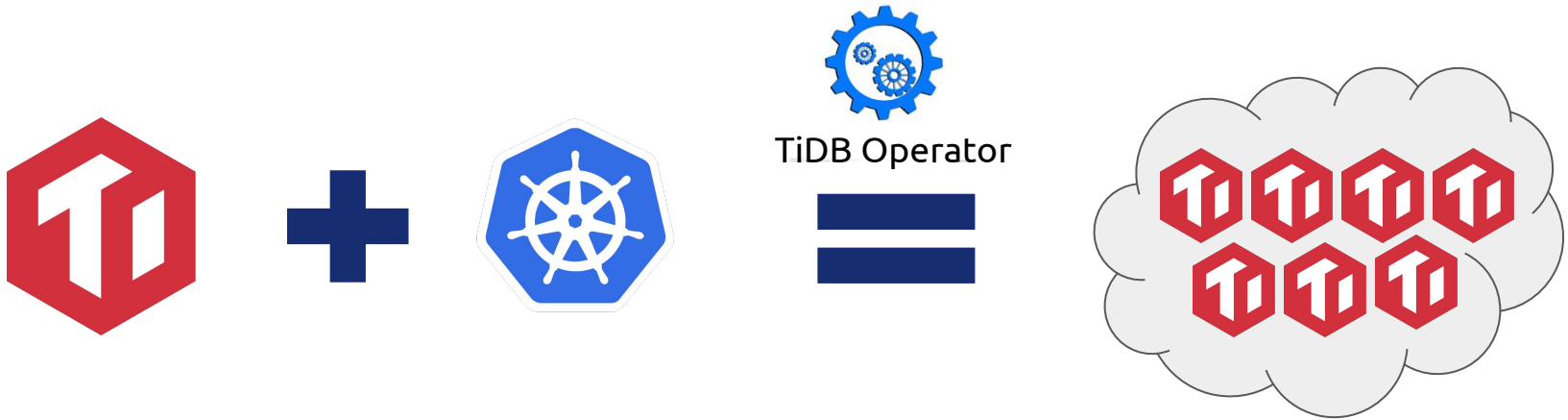



# Kubernetes: powerful extensibility

- Standard interface: CNI, CRI, CSI
- Scheduler: scheduler extender
- Controller: CRD
- APIServer: Aggregated APIServer
- Kubelet: virtual kubelet
- Cloud Provider: LoadBalancer, PersistentVolume
- ...

# Part III - How we make it possible

# TiDB Operator

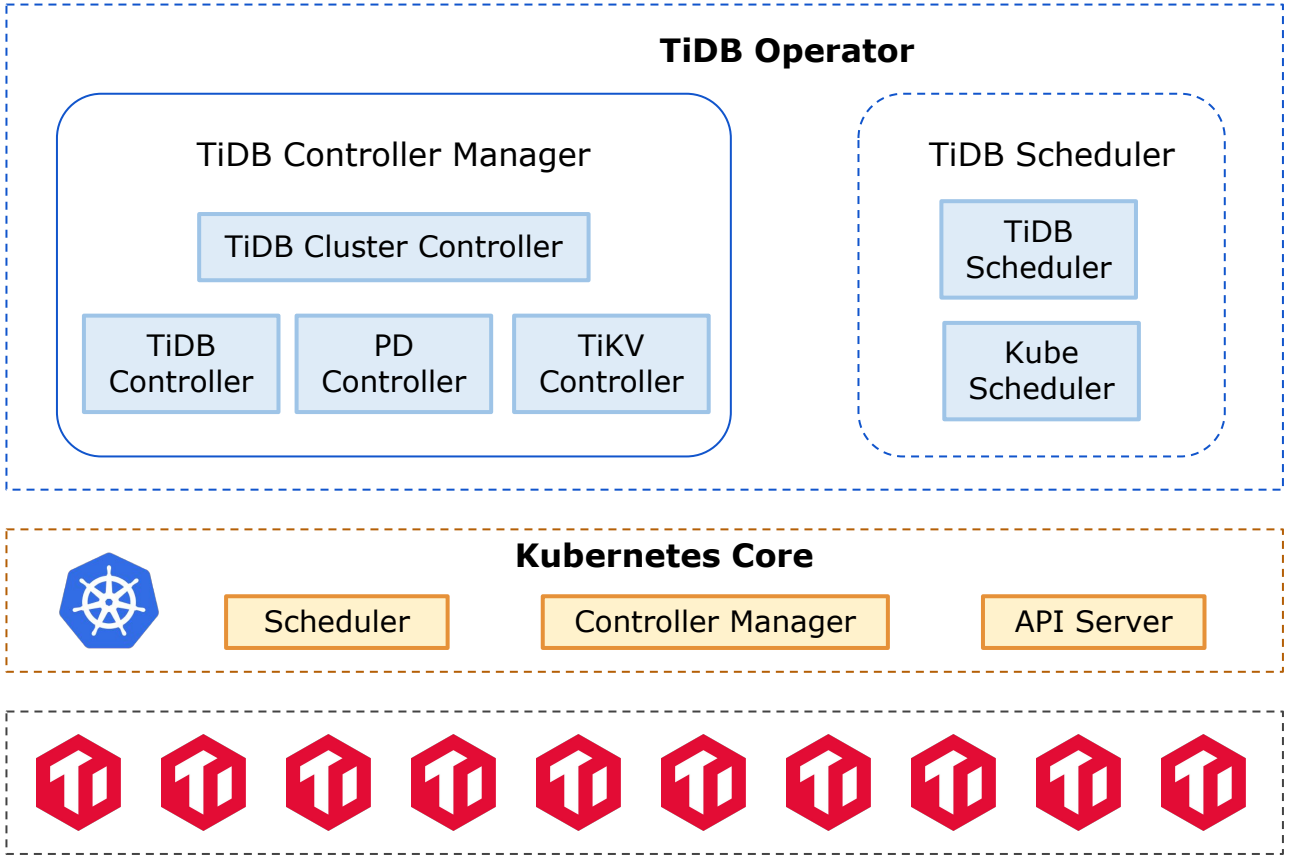



https://github.com/pingcap/tidb-operator

# Features

- Manage multiple TiDB clusters
- Safely scale the TiDB cluster
- Easily installed with Helm charts
- Network/Local PV support
- Automatically monitoring the TiDB cluster
- Seamlessly perform rolling updates to the TiDB cluster
- Automatic failover
- TiDB related tools integration

# Architecture



# How we manage state

Kubernetes builtin controllers

Deployment:

- Start ✓
- Scale ✓
- Upgrade ✓
- Failover ✓

StatefulSet:

- Start √≠
- Scale √≠
- Upgrade √≠
- Failover ×

# TiDB Operation

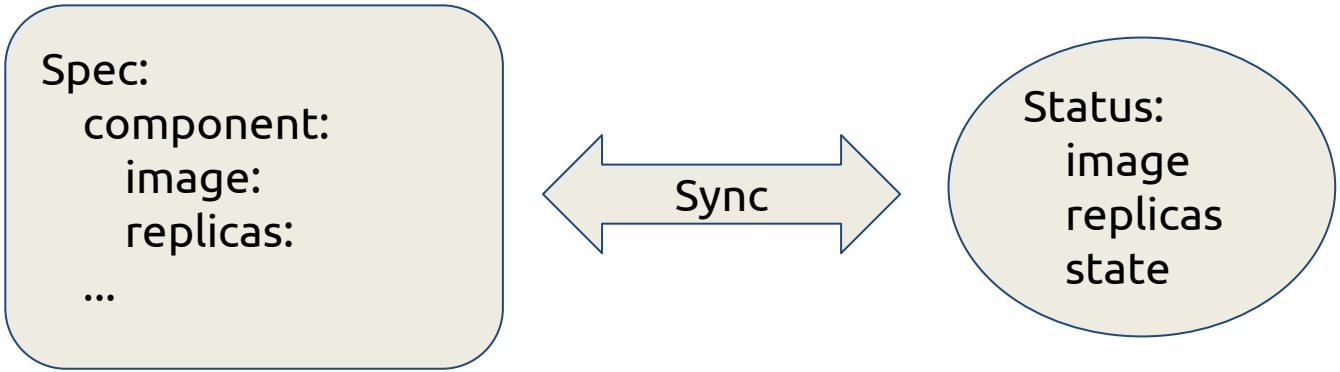
- Cluster bootstrap: Initial PD -> New PD joins existing cluster
- Safely delete PD
  - remove member using PD API
  - stop pd-server
- Safely delete TiKV
  - offline store using PD API
  - stop tikv-server
- Graceful upgrade
  - PD: transfer Raft leader
  - TiKV: evict Raft leaders
  - TiDB: evict DDL owner

# Custom controller

Domain operation logic

- ThirdPartyResource (TPR), CustomResourceDefinition (CRD):
  - Simple & easy
  - Lack schema & versionning (added in newer version)
- Aggregated APIServer (AA):
  - Powerful but complicated
  - Coupled with the built-in APIServer, hard to deploy

# Custom controller



# Custom controller

```
type Manager interface {
    Sync(*TidbCluster) error
}
```

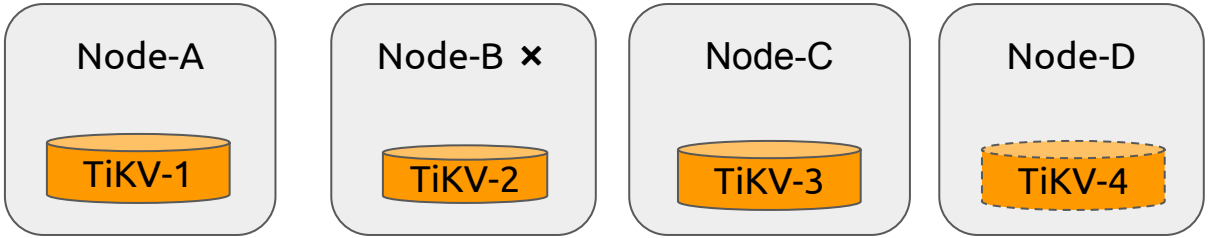
```
apiVersion: pingcap.com/v1alpha1
kind: TidbCluster
metadata:
  name: demo
spec:
  pd:
    image: pingcap/pd:v2.1.0
    replicas: 3
    requests:
      cpu: “4”
      memory: “8Gi”
…
  tikv:
    image: pingcap/tikv:v2.1.0
```

```
…
status:
  tikv:
    stores:
      “5”:
        podName: demo-tikv-2
        state: Up
…
```

# Custom controller

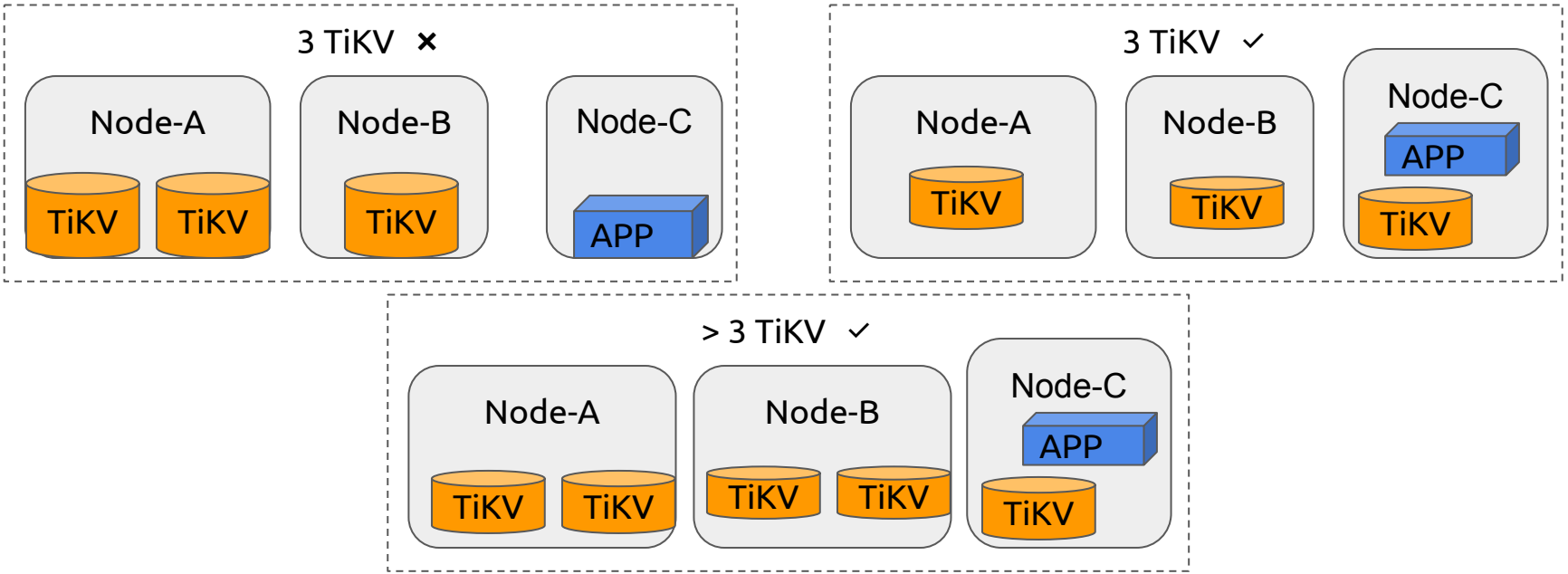
StatefulSet with Local PV failover:

1. Increase replicas when failure occurs
2. Decrease replicas when node come back (ordinal limitations with statefulset)



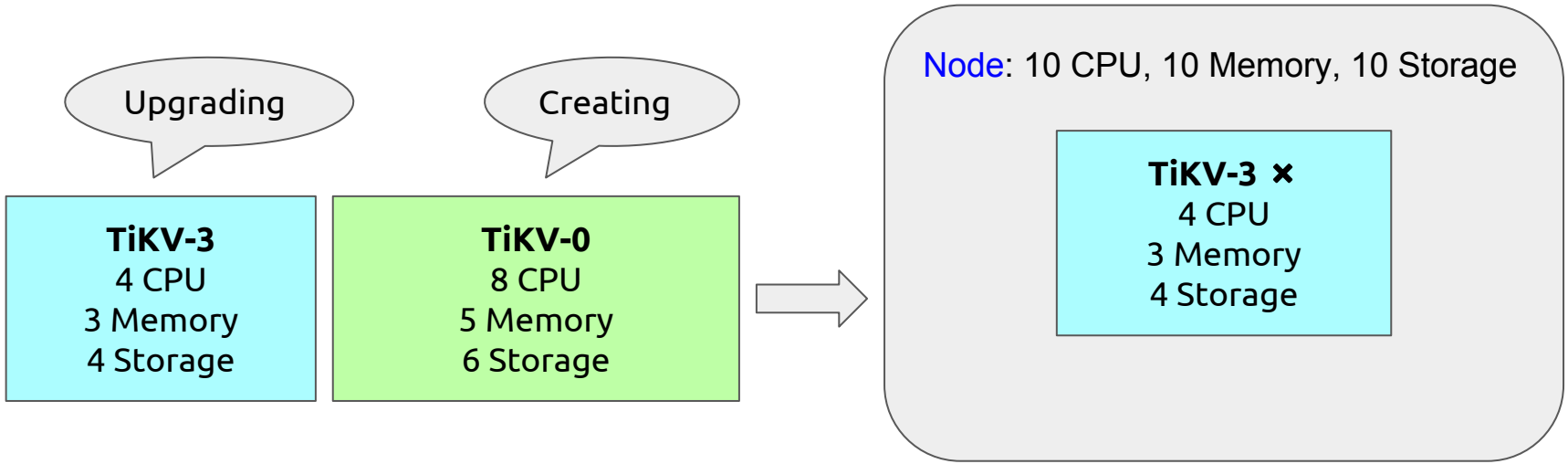
# How we schedule stateful app

- Schedule consider existing pods topology



# How we schedule stateful app

- Schedule consider *virtual* resource for local volume



# TiDB Operator

Open-sourced! ヾ(=^▽^=)ノ🎉

https://github.com/pingcap/tidb-operator

# THANKS