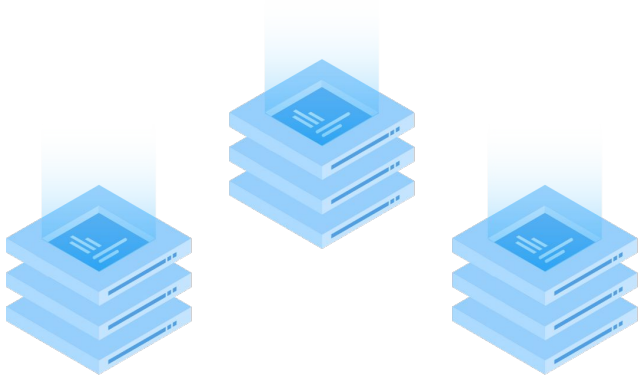
Distributed Database on Cloud

Presented by Hailong Zhang

PingCAP TiDB



Agenda

- Intro to TiDB
- Kubernetes Storage for TiDB
- TiDB on Kubernetes
- TiDB on Public Cloud
- Challenges

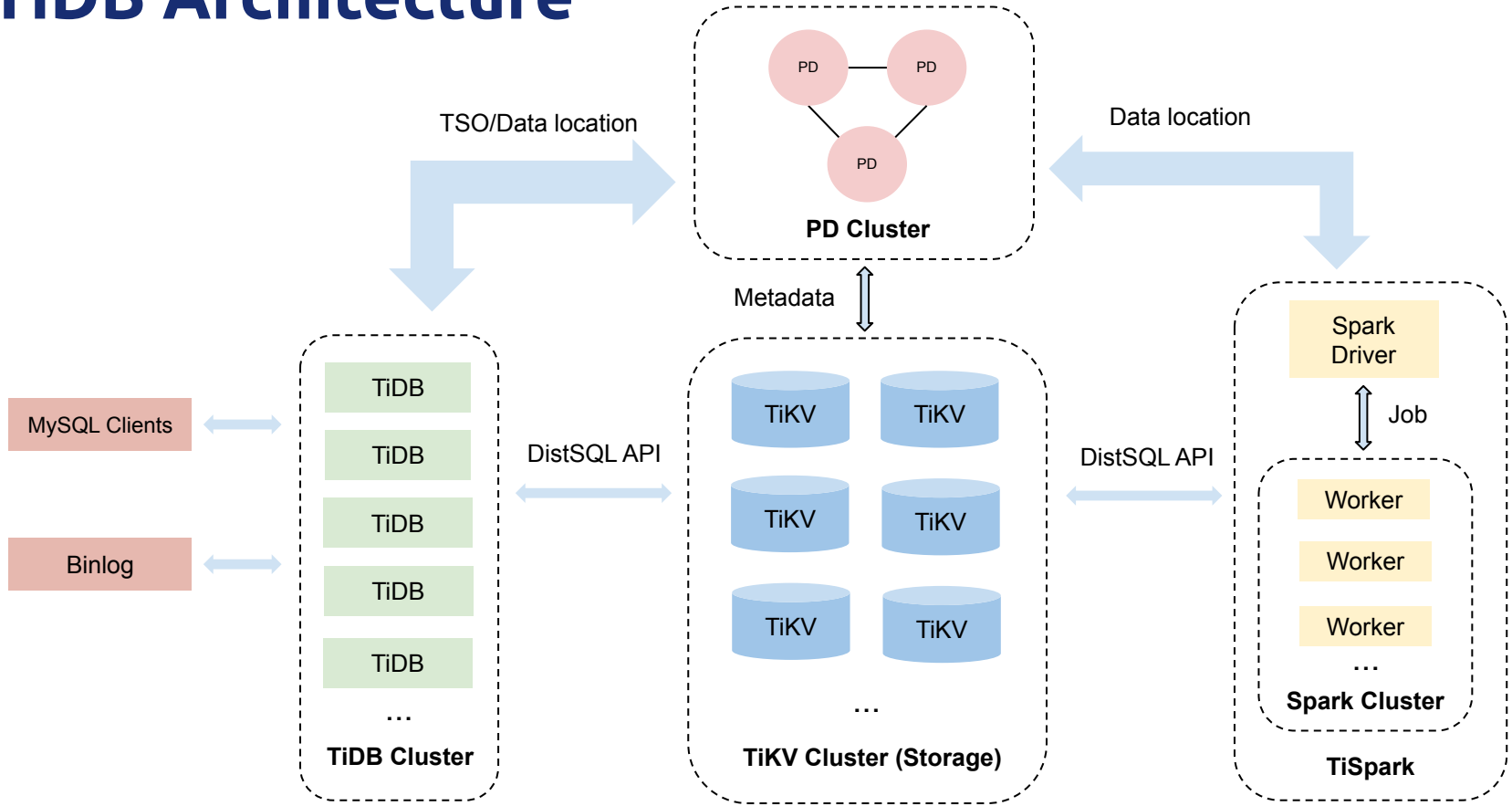
PingCAP

Part I - Intro to TiDB

What is TiDB?

- An **open-source** distributed **NewSQL** database for hybrid transactional and analytical processing (**HTAP**) which speaks MySQL procotol

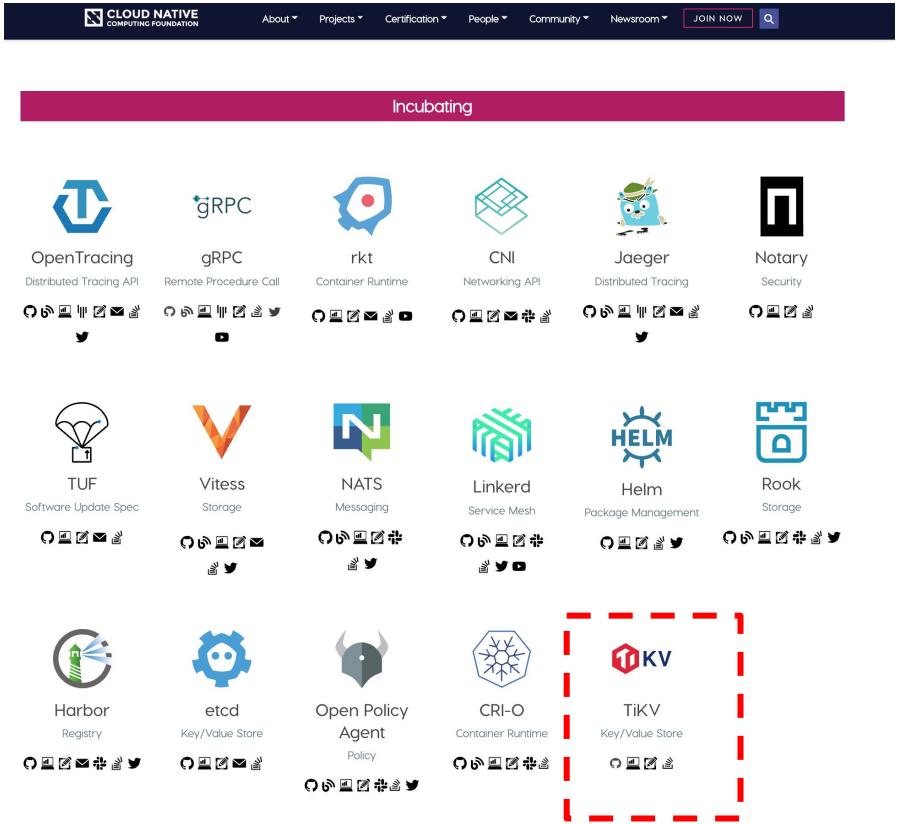
TiDB Architecture



Cloud Native DataBase

Cloud native technologies empower organizations to build and run **scalable** applications in modern, dynamic environments such as public, private, and hybrid **clouds**.
Containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this approach.

---- CNCF

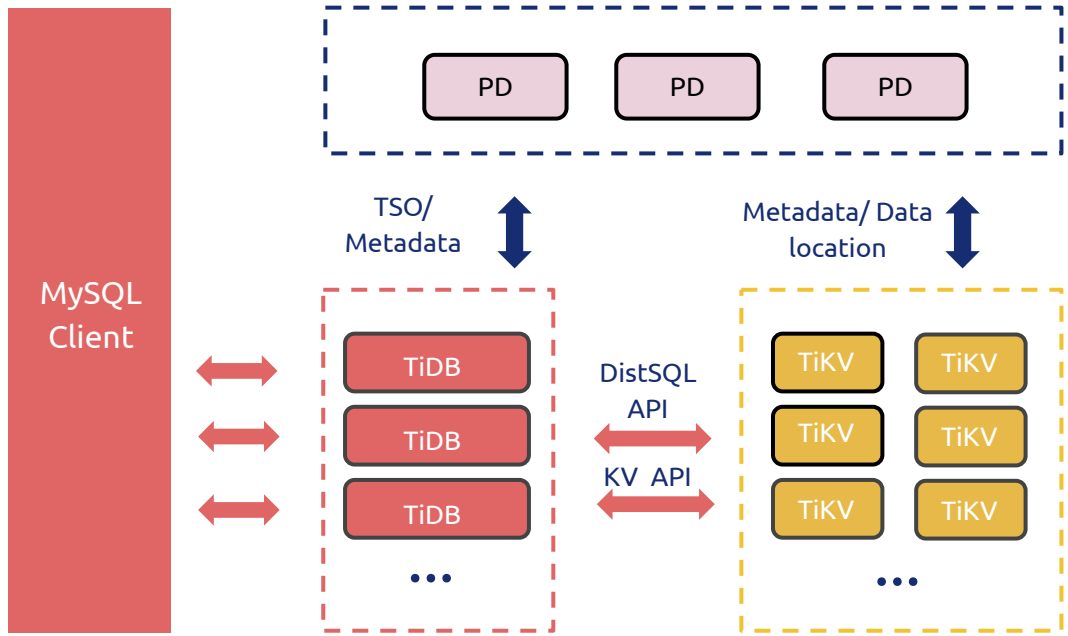


PingCAP

Part II - Kubernetes Storage for TiDB

TiDB Key Components

- TiDB
  - CPU Intensive
  - Stateless
- TiKV
  - CPU and I/O Intensive
  - Stateful
    - Unique network identifiers
    - Persistent storage
- PD
  - Lightweight
  - Stateful



Kubernetes Storage

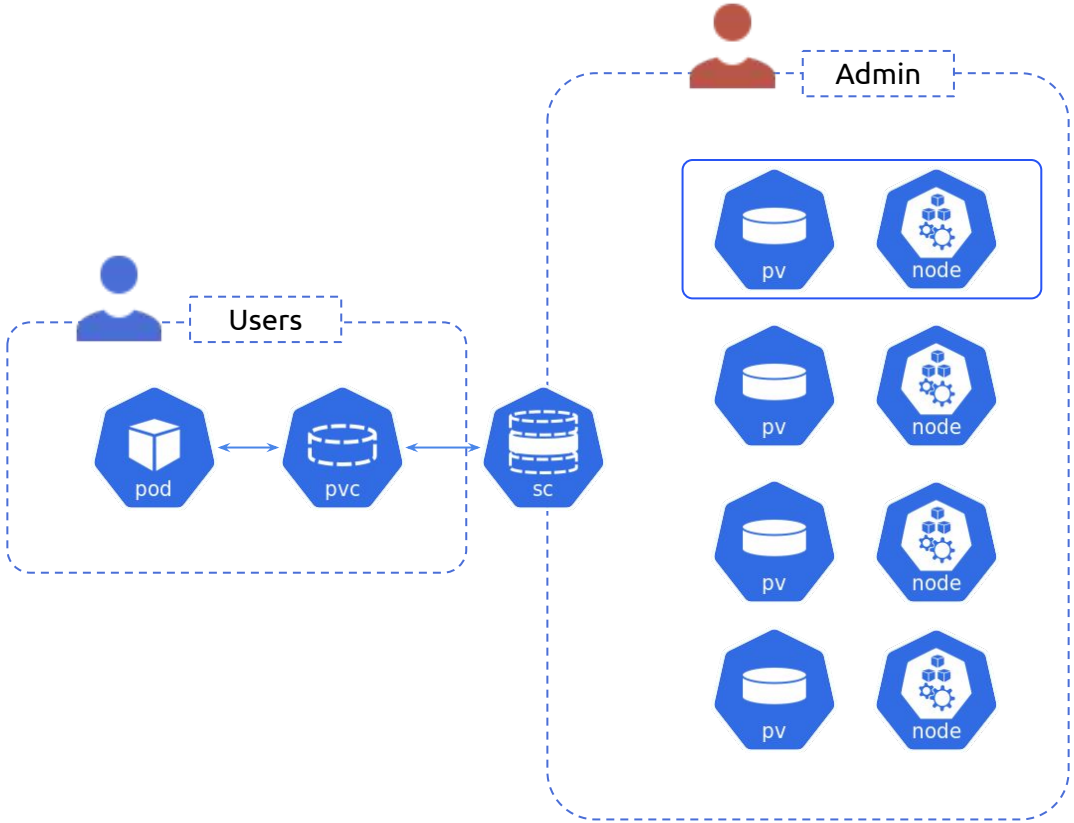
Type	Lifecycle	Use Case
Local Ephemeral Storage	Pod	EmptyDir, Secret...
Remote Persistent Storage	Independent of Cluster	Ceph, Cloud Persistent Disk, NFS...
Local Persistent Storage	Disk or Node	Local PV, Hostpath

Type	IOPS	Throughput	Latency	Capacity	Durability
Remote (Networked) Storage	Low	Low	High	High	Yes
Local Storage	High	High	Low	Low	No

Kubernetes Storage - Local PV VS. HostPath

Type	Reference	Scheduler Aware	Block Device	Use Cases
Hostpath	PVC or Directly	No	No formatting	• Mount/proc into node_exporter
Local PV	PVC	Yes	Support formatting	• Distributed systems which provide fault tolerance in case of node failures, e.g. PD, TiKV • Cache systems which tolerate data loss can avoid data rebuilding on pod restart, e.g. CDN frontend

Kubernetes Storage - Local PV



Kubernetes Storage - Local PV

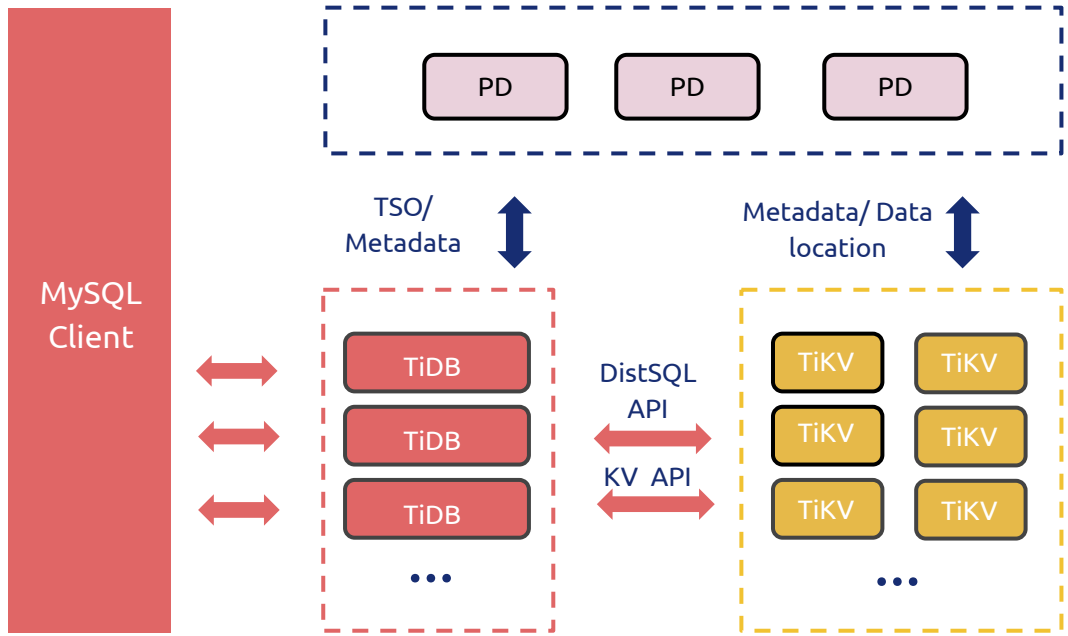
- 1.14 GA
- Local Volume Provisioner
  - https://github.com/kubernetes-sigs/sig-storage-local-static-provisioner
- Best Practice
  - IO Isolation
    - a whole disk per volume
  - Capacity Isolation
    - separate partitions per volume
  - Avoid recreating nodes with the same node name
  - Utilize UUID in mount point for volumes with a filesystem
  - Use a unique ID for raw block volumes
  - persistentVolumeReclaimPolicy: Retain (if necessary)

PingCAP

Part III - TiDB on Kubernetes

TiDB Key Components

- TiDB
  - CPU Intensive
  - Stateless
- TiKV
  - CPU and I/O Intensive
  - Stateful
    - Unique network identifiers
    - Persistent storage
- PD
  - Lightweight
  - Stateful



Kubernetes Resource

- TiDB
  - Deployment
- TiKV
  - Statefulset
- PD
  - Statefulset

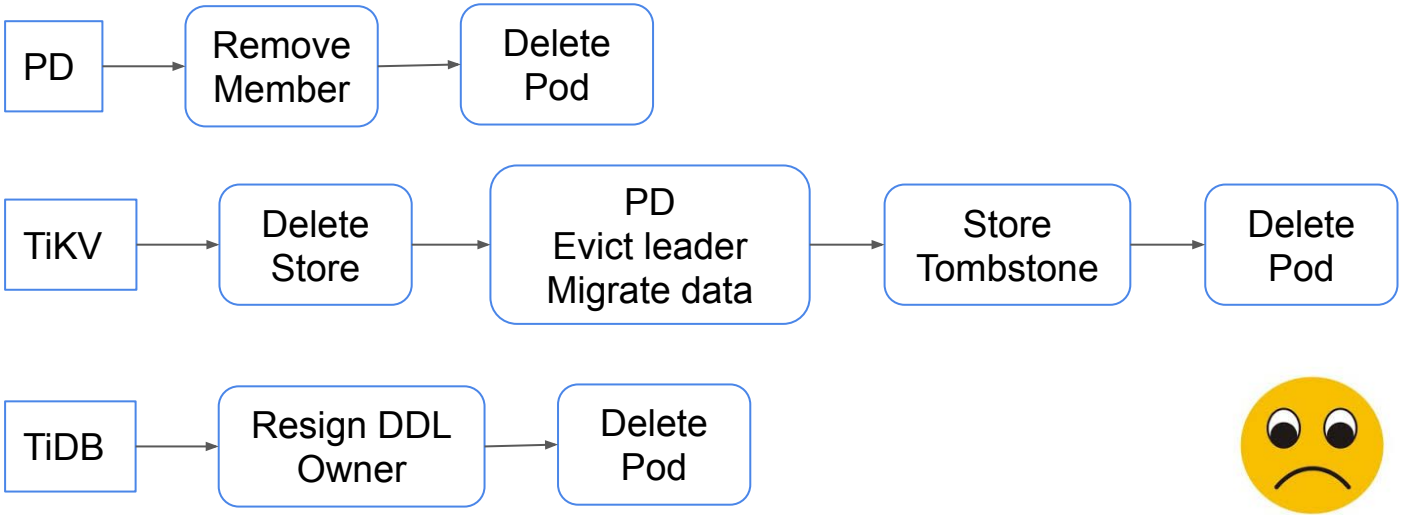
- DDL Owner
- Other features

Statefulset

Statefulset for all key components

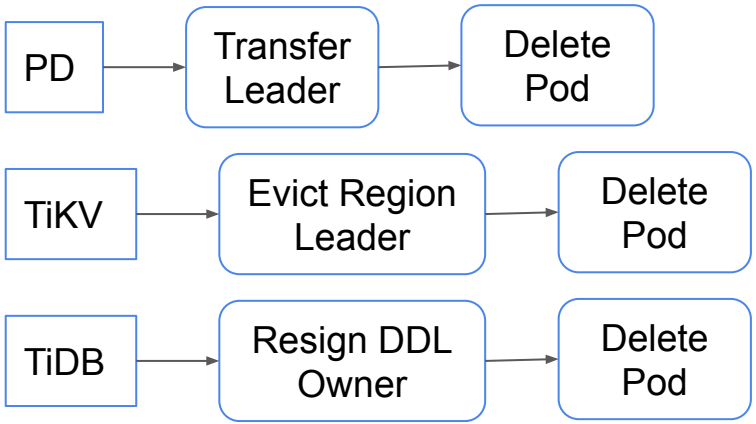
Statefulset - Scale In

Kubernetes Application Scale In: decrease replicas -> controller deletes Pod



Statefulset - Rolling Update

- Rolling update:
  - Cluster version
  - Cluster configuration
- StatefulSet can perform rolling update out-of-box



Statefulset - Failover

- Failover
  - Containers down
  - Node down
    - Physical down
    - Network Partition

- No Failover for Statefulset Pods

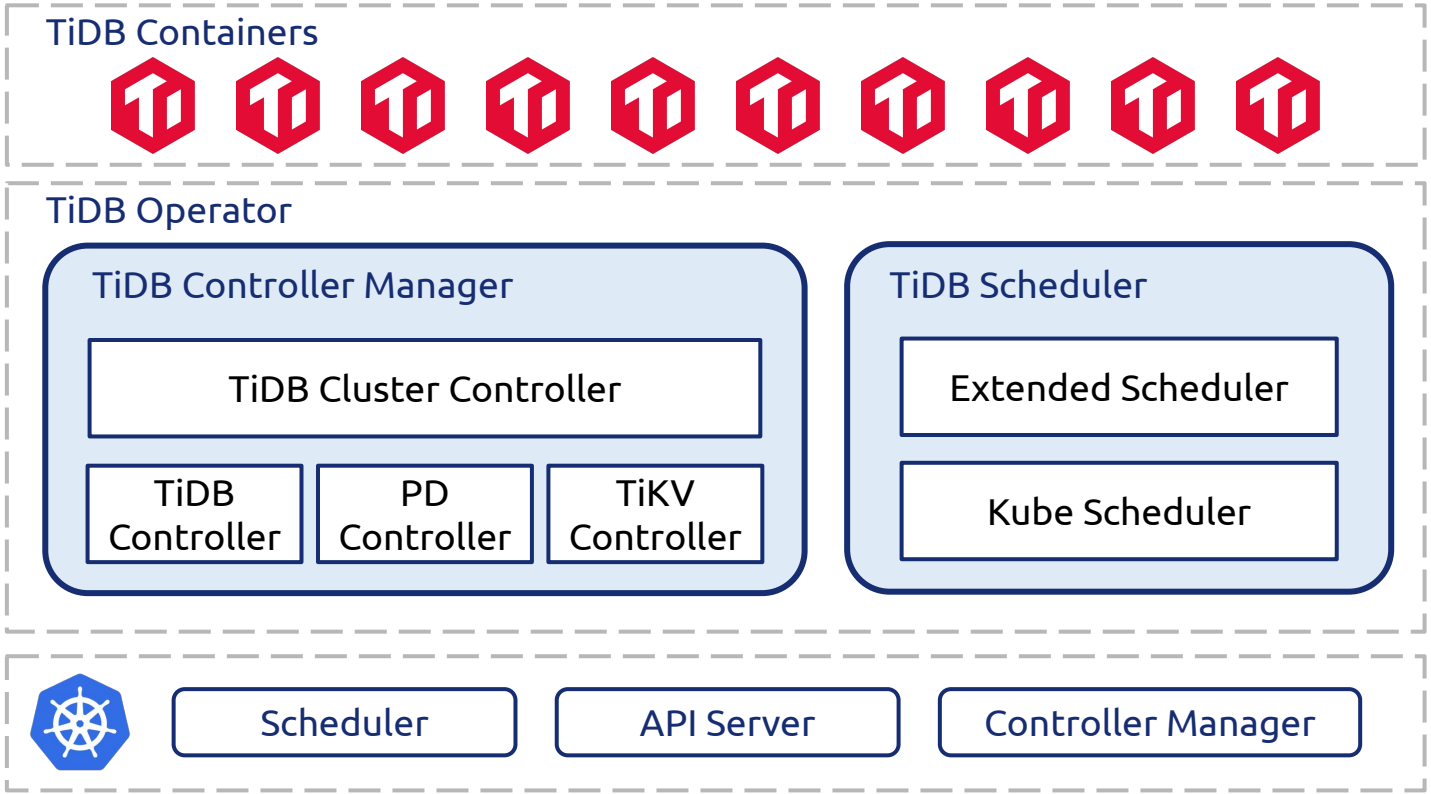
Solution

- Any other resources?
- Kubernetes with private code update
  - Upstream update
  - Managed kubernetes
- Extend Kubernetes - Operator
  - Custom Resource
  - Custom Controller
  - Custom Scheduler

TiDB Operator

- Kubernetes as the orchestration platform
- TiDB Operator injects TiDB's domain-specific orchestration logic into Kubernetes:
  - **TidbCluster**: the custom resource to declare user's intention
  - **tidb-controller-manager**: a set of custom controllers that implements the user' intention declared in **TidbCluster**
  - **tidb-scheduler**: custom scheduling policy, e.g. PD and TiKV HA(High Available) scheduling

TiDB Operator Architecture



TiDB Operator CRD

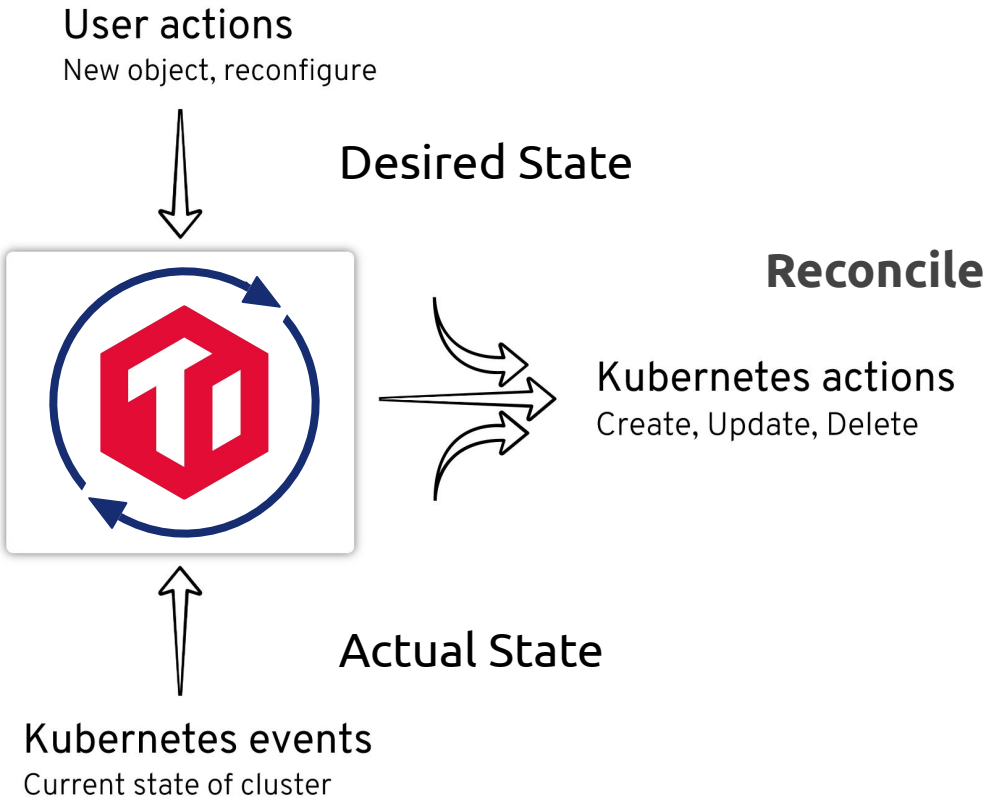
```yaml
apiVersion: apiextensions.k8s.io/v1beta1
kind: CustomResourceDefinition
metadata:
  name: tidbclusters.pingcap.com
spec:
  group: pingcap.com
  scope: Namespaced
  names:
    plural: tidbclusters
    singular: tidbcluster
    kind: TidbCluster
    shortNames:
    - tc
  validation:
    openAPIV3Schema:
```

Custom Resource Definition

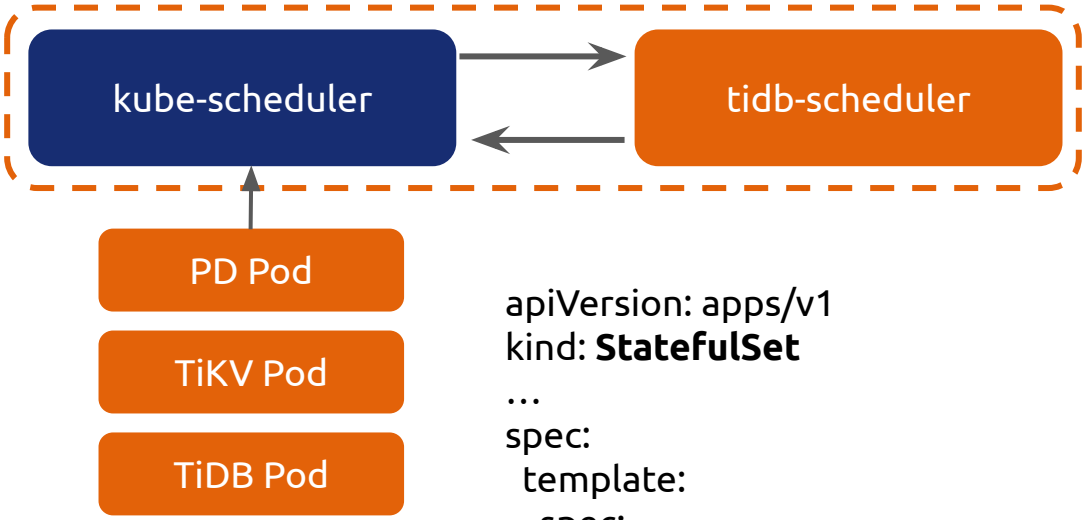
```yaml
kind: TidbCluster
metadata:
  name: aylei-tidb
spec:
  schedulerName: tidb-scheduler
  pd:
    image: pingcap/pd:v2.1.0
    replicas: 3
  tidb:
    image: pingcap/tidb:v2.1.0
    maxFailoverCount: 3
    replicas: 4
  tikv:
    image: pingcap/tikv:v2.1.0
    replicas: 5
```

Custom Resource

TiDB Controller Manager



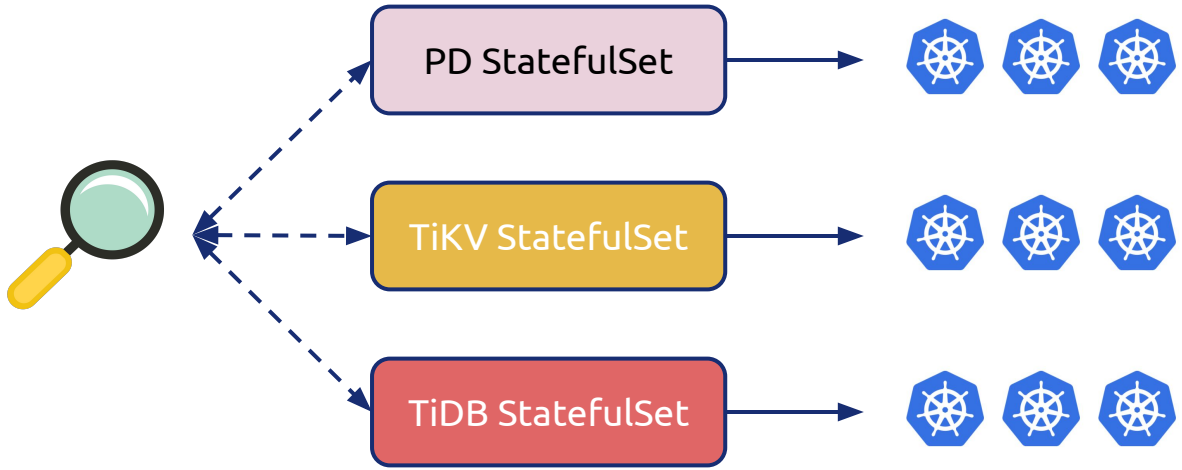
Extended Scheduler




```
apiVersion: apps/v1
kind: StatefulSet
…
spec:
  template:
    spec:
      schedulerName: tidb-scheduler
      containers:
...
```

TiDB Operator

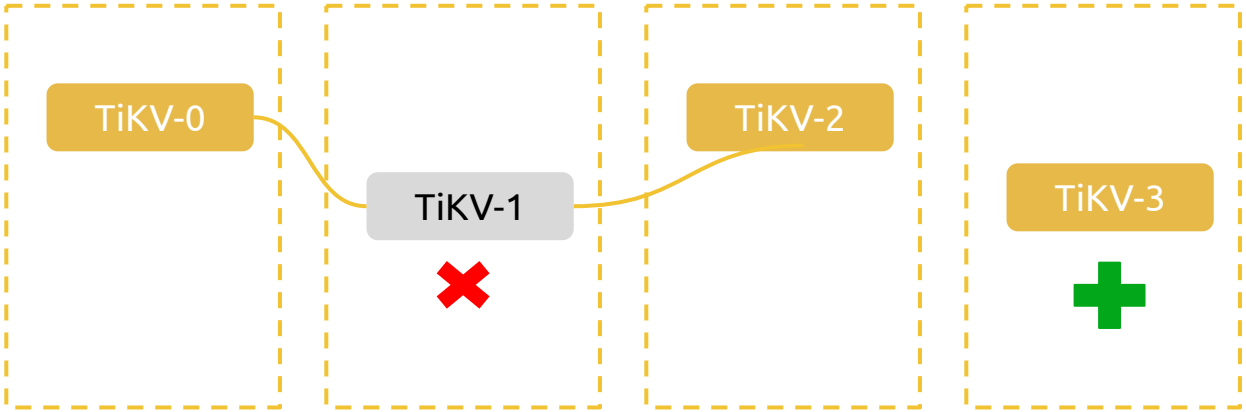
```
apiVersion: pingcap.com/v1alpha1
kind: TidbCluster
metadata:
  name: demo
spec:
  pd:
    image: pingcap/pd:v2.1.3
    replicas: 3
    ...
  tikv:
    image: pingcap/tikv:v2.1.3
    replicas: 5
    ...
  tidb:
    image: pingcap/tidb:v2.1.3
    replicas: 2
    ...
```



TiDB Operator Features

- Bootstrap and 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

Auto Failover




```
status:
   tikv:
     failureStores:
       instance: TiKV-1
```

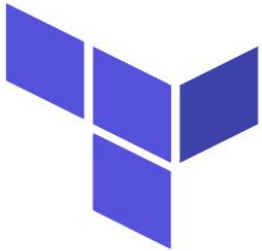
Happy Ending

Custom Resource + TiDB Operator + Local PV + Raft

PingCAP

Part IV - TiDB on Public Cloud

Public Cloud



HashiCorp

Terraform

Write, Plan, and Create Infrastructure as Code

Amazon EKS

Google Kubernetes Engine

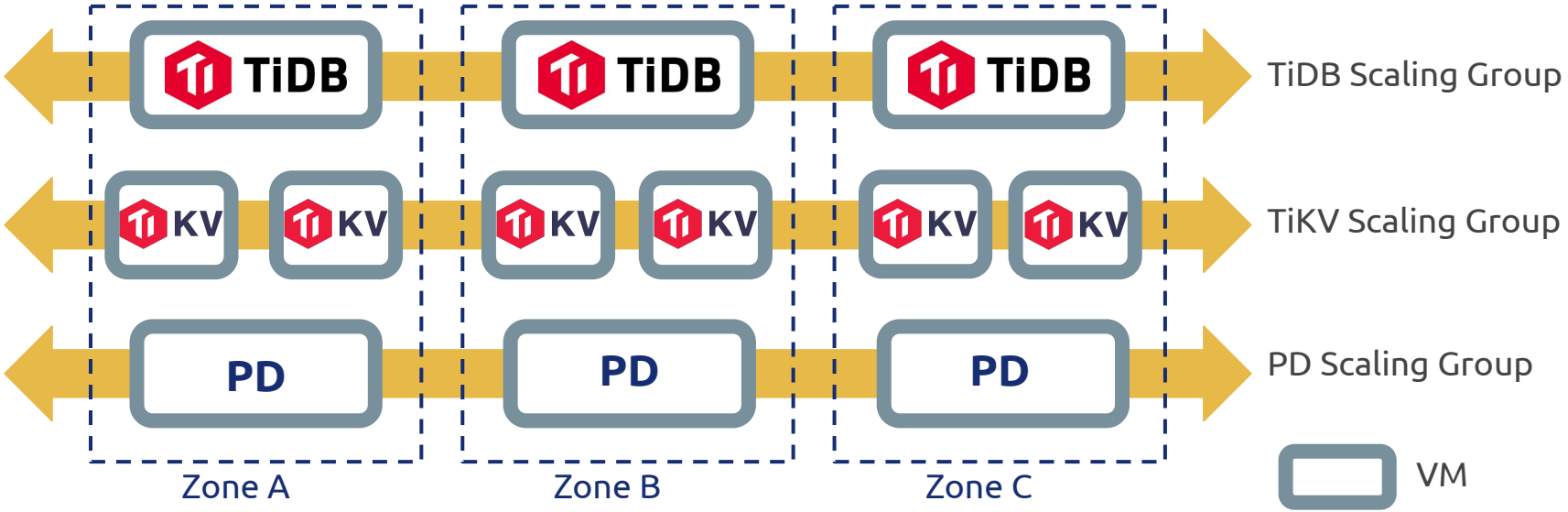
Alibaba ACK

Public Cloud

- Terraform Modules
  - VPC, bastion, etc.
  - tidb-operator
    - Kubernetes cluster
    - Auto scaling group for TiDB Operator
    - TiDB Operator
  - tidb-cluster
    - Auto scaling groups for PD, TiKV, TiDB and monitor
    - TiDB Cluster
- Internal LB for TiDB Service
- External LB for Monitor Service
- Local SSD
- Multiple TiDB Clusters
- Multiple Kubernetes Clusters

Public Cloud

- Cloud TiDB recommends using dedicated node for PD/TiKV/TiDB in production environment



Happy Ending

Custom Resource + TiDB Operator + Local PV + Raft + Terraform

PingCAP

Part V - Challenges

Challenges

Statefulset
+ Local PV

User: We want to take one node offline.
Us: Can you take the node offline where the TiKV pod with largest ordinal is scheduled…
User: What...

Custom Statefulset
+
Custom Sts Controller

User: Upgrade failed…
Us: Uncompatible configuration change from xx…
It's time to change...

Aggregated API Server

TEP: Helm client is too poor to integrate with TEP…
Why you do uncompatible changes to helm values?
Us: ...

Challenges on GKE

- Local SSD
  - Only SCSI interface (slow), NVMe is in early alpha status
  - Disk size is limited (375GB)
    - Combining disk is unsupported and node restart may break
- Instance group failover creating a new node with the same node name causing TiKV pod crash (data lost)
- GKE worker node upgrade is not graceful
  - Must disable automatic repair and upgrade
- GKE masters auto-upgrade
- GKE regional cluster forces worker instances created evenly on all AZs. Introduce extra cost for monitor and control plane.

Challenges on EKS

- VM in cross AZ auto scaling group is not guaranteed to be created in the expected AZ
- Data lost for deleted unhealthy VM in auto scaling group
- k8s version cannot upgrade

Your Contribution is Welcome!

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

PingCAP

Thank You!

Any Questions ?

关注 PingCAP 官方微信

了解更多技术干货