Go 夜读 & 云原生社区

TiDB Operator 架构与实现

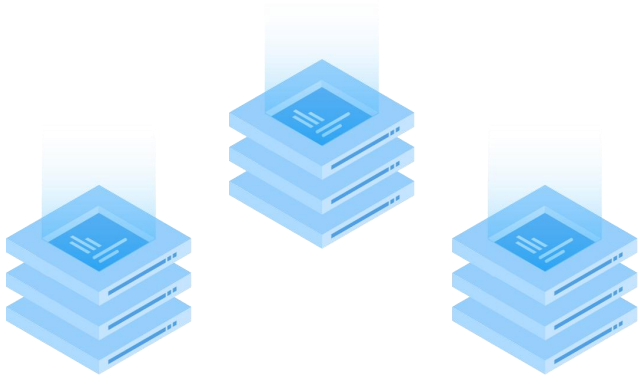
扫描下方二维码

填写你的问题

TiDB Operator: Design & Implemention

Presented by Yecheng Fu (@cofyc)

PingCAP TiDB



Agenda

- TiDB Operator 简介
- 扩展 Kubernetes 的几种方式
- TiDB Operator 实现

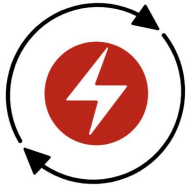
PingCAP

TiDB Operator 是什么

Cloud Native Era: Portable, Scalable, Automated

TiDB

TiDB Operator

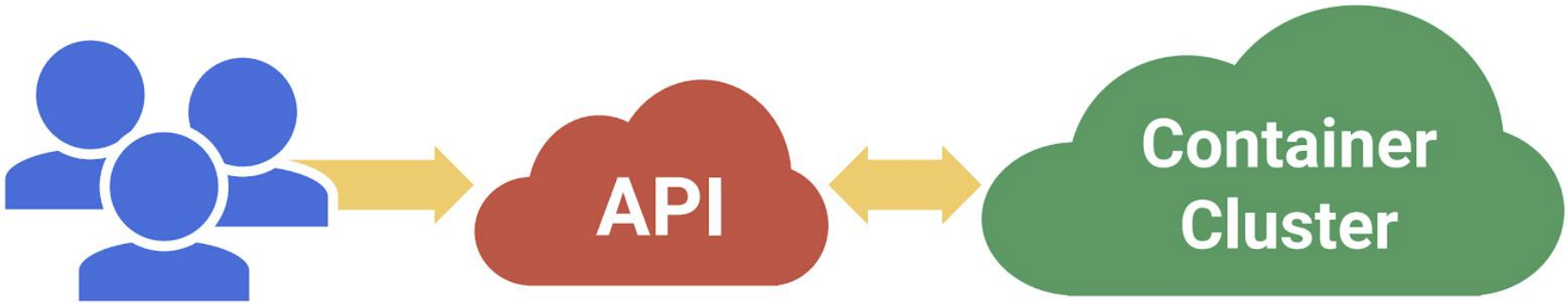


Full lifecycle management of TiDB cluster

- Deployment
- Upgrading
- Scaling
- Handle network, hardware failures, etc.
- Backup/Restore/Data migration
- ...

阿里云 aws Google Cloud Microsoft Azure IDC

Kubernetes Pattern - Declarative Model



- 用户描述自己的期望，提交给 Kubernetes API Server
- Kubernetes 根据用户的期望以及当前的状态，协调各方达成用户的期望

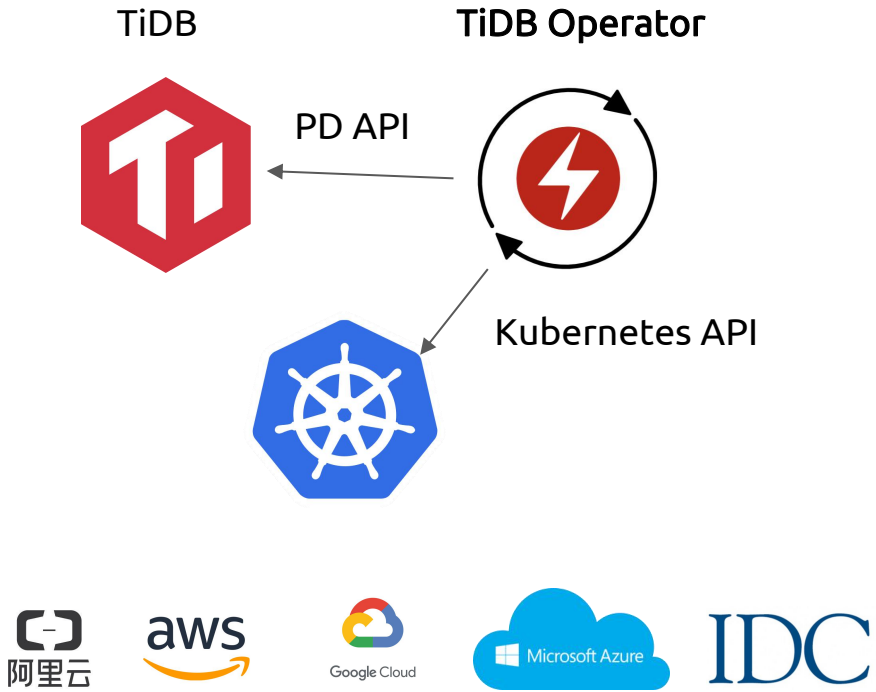
Kubernetes Pattern - Declarative Model

- Kubernetes has pod, deployment, statefulset, etc.
- But it does not know how to operate a TiDB cluster...

Kubernetes Pattern - Declarative Model

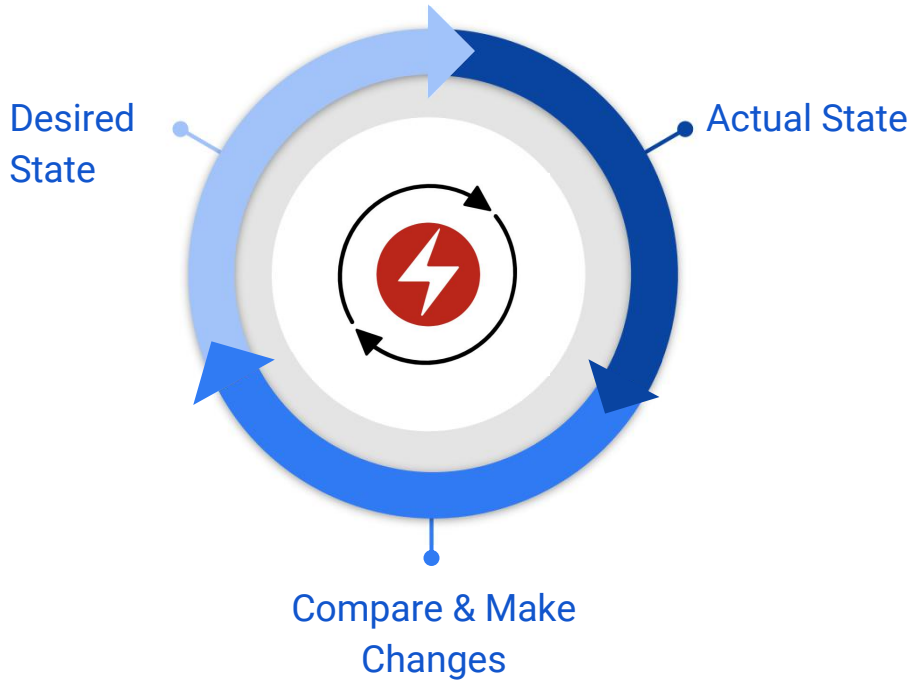
- Kubernetes has pod, deployment, statefulset, etc.
- But it does not know how to operate TiDB cluster…
- Until we implemented TiDB Operator

Operator Pattern - Extending Kubernetes



Operator Pattern - Custom Resource & Controller

```yaml
apiVersion: pingcap.com/v1alpha1
kind: TidbCluster
spec:
  version: v4.0.3
  pd:
    baseImage: pingcap/pd
    replicas: 3
    requests:
      storage: "1Gi"
    ...
  tikv:
    <desired state of TiKV>
  tidb:
    <desired state of TiDB>
status:
  pd:
    <observed state of PD>
  tikv:
    <observed state of TiKV>
  tidb:
    <observed state of TiDB>
```



PingCAP

扩展 Kubernetes 的几种方式

Common ways to extend Kubernetes

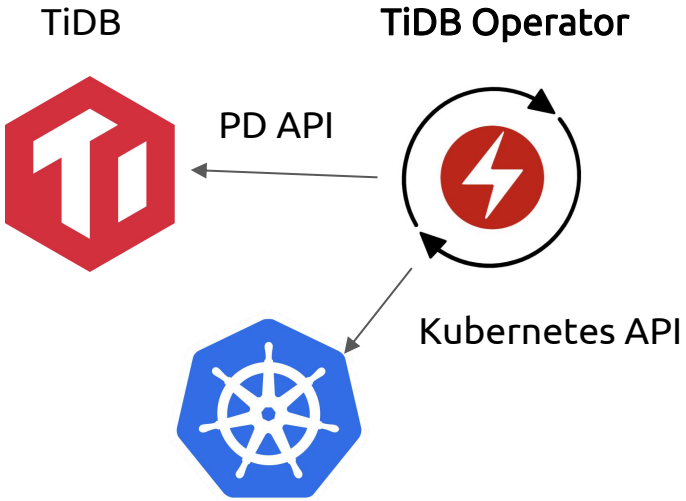
- Custom Resource
  - TidbCluster
  - TidbInitializer
  - TidbMonitor
  - Backup/Restore
- Custom Controller
- Scheduler Extender (optional)
- Admission Webhook (optional)
- …Scheduler framework, Aggregated APIServer

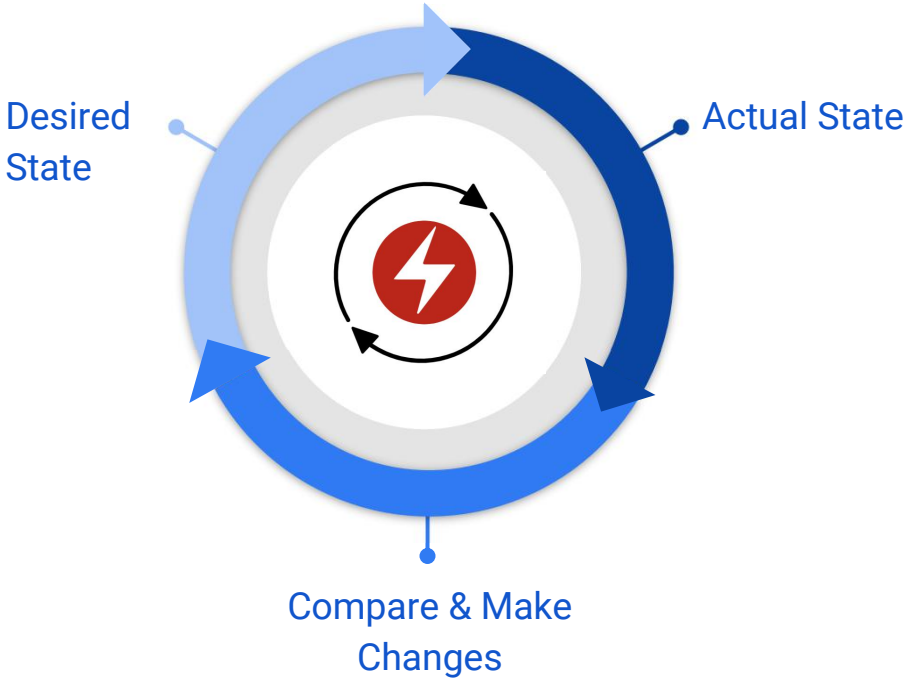
Custom Resource

```yaml
apiVersion: pingcap.com/v1alpha1
kind: TidbCluster
spec:
  version: v4.0.3
  pd:
    baseImage: pingcap/pd
    replicas: 3
    requests:
      storage: "1Gi"
    ...
  tikv:
    <desired state of TiKV>
  tidb:
    <desired state of TiDB>
status:
  pd:
    <observed state of PD>
  tikv:
    <observed state of TiKV>
  tidb:
    <observed state of TiDB>
```

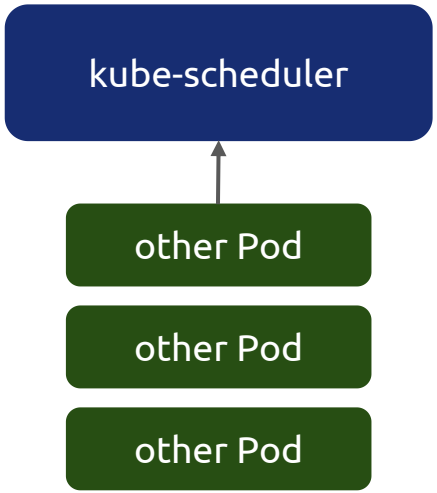
```yaml
kind: CustomResourceDefinition
spec:
  group: pingcap.com
  names:
    kind: TidbCluster
    plural: tidbclusters
    shortNames:
    - tc
  validation:
    openAPIV3Schema:
     type: object
     properties:
        spec:
          type: object
          properties:
            pd:
              type: object
            tikv:
              type: object
            tidb:
              type: object
```

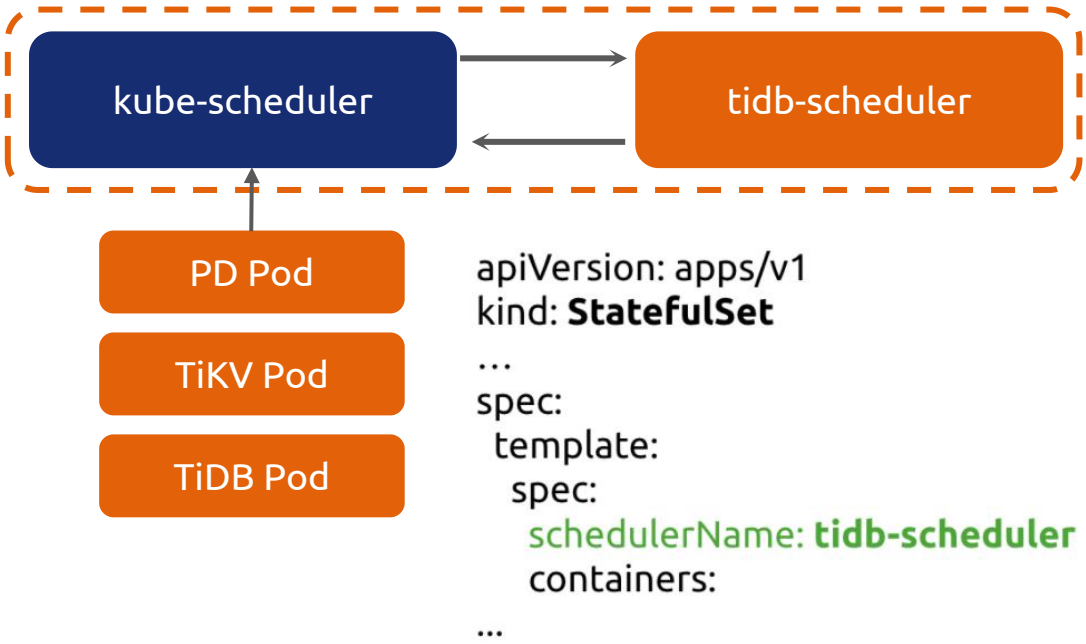
Custom Controller



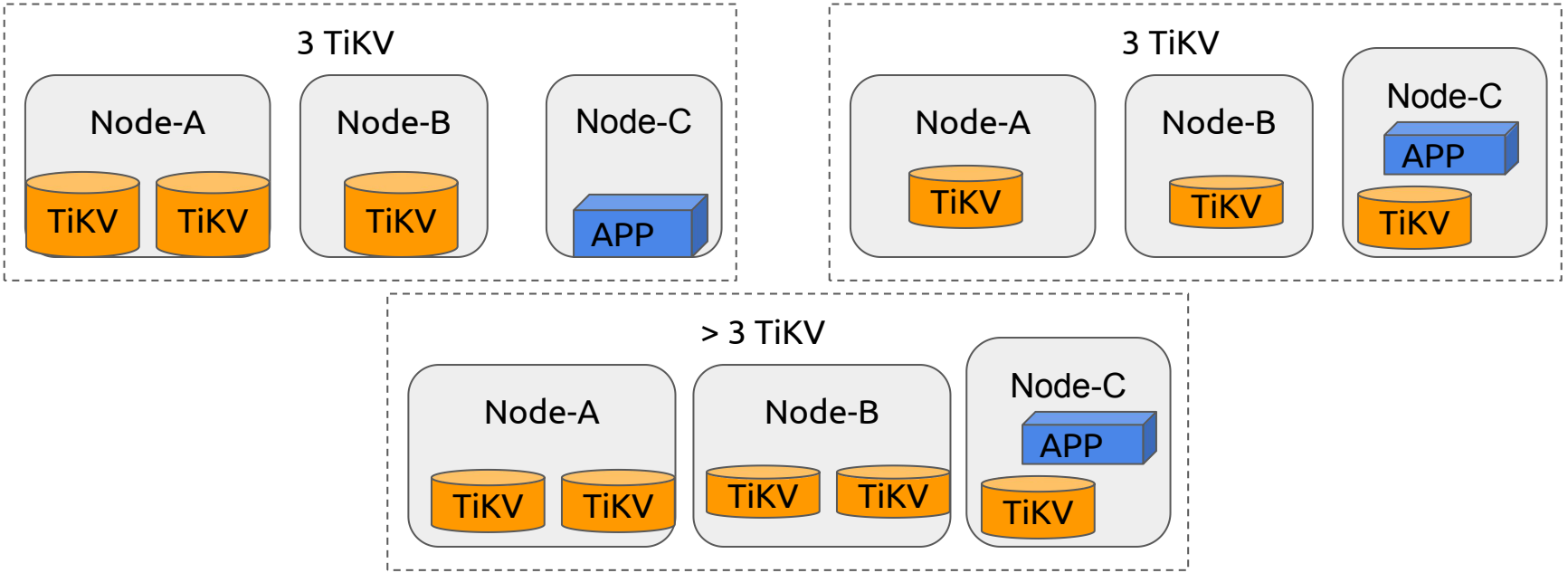


Scheduler Extender



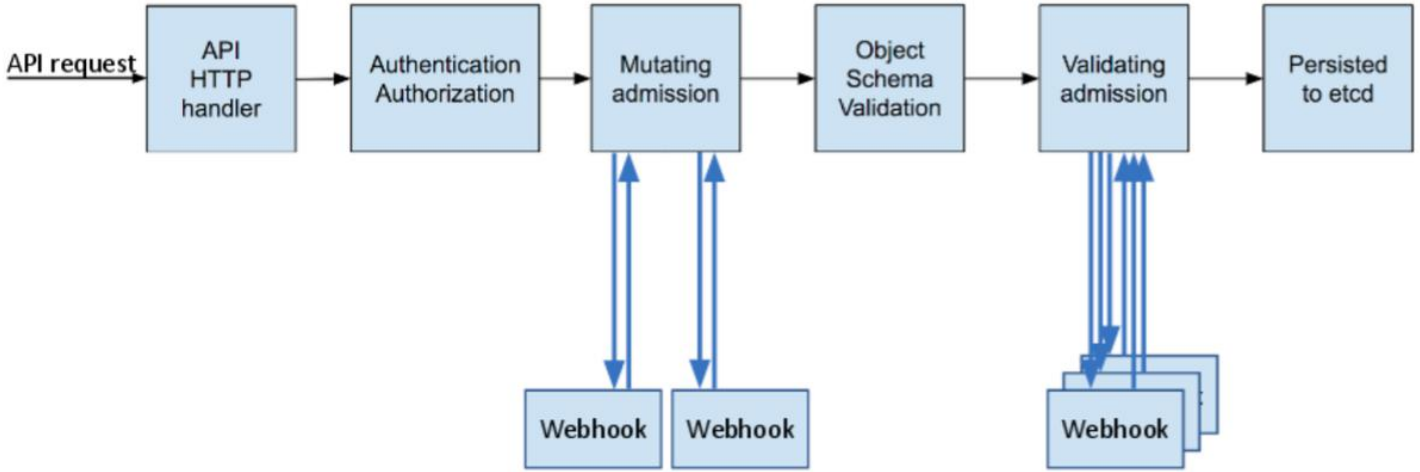


Scheduler Extender



* *EvenPodsSpread* feature introduced in Kubernetes 1.18 provides a more flexible solution

Admission Webhook



Able to intercept API requests and may change or deny it.

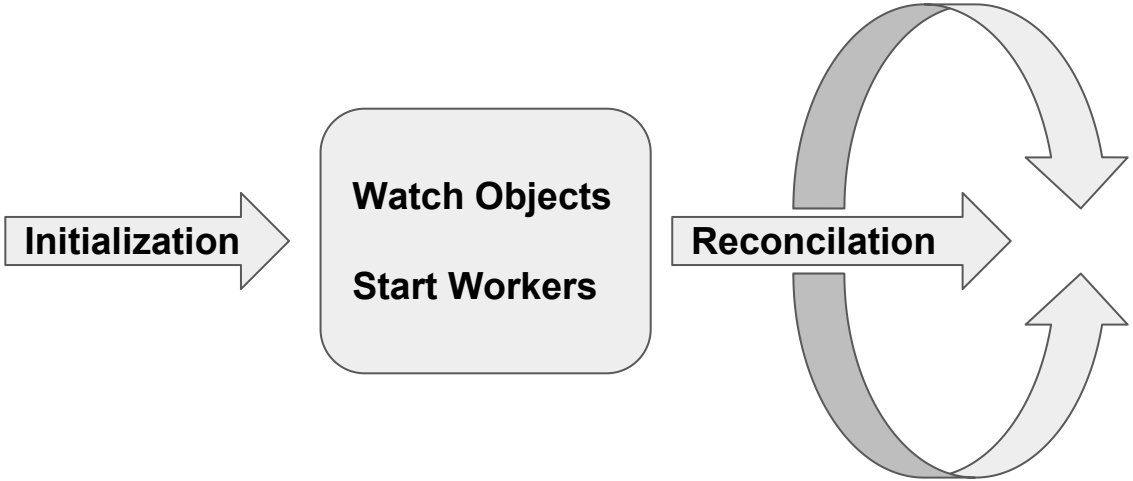
PingCAP

TiDB Operator 实现

TiDB Operator Implementation

- The tidb-controller-manager
  - Initialization
  - Watch objects (Tidb Cluster CR)
  - Reconcilation
    - on change of watched CR objects
    - periodically for TiDB clusters (30s by default)
- Full lifecycle management of a TiDB Cluster

tidb-controller-manager - Flowchart



tidb-controller-manager - Initialization

```
// Create informer factories for custom and Kubernetes
informerFactory := informers.NewSharedInformerFactoryWithOptions(cli, resyncDuration)
kubeInformerFactory = kubeinformers.NewSharedInformerFactoryWithOptions(kubeCli, resyncDuration)
// Create controllers
tcContrller := tidbcluster.NewController(cli, kubeCli, informerFactory, kubeInformerFactory)
...
// Start informer factories after all controller are initialized
informerFactory.Start(ctx.Done())
kubeInformerFactory.Start(ctx.Done())
// Wait for all started informers' cache were synced
informerFactory.WaitForCacheSync(wait.NeverStop)
kubeInformerFactory.WaitForCacheSync(wait.NeverStop)
// Start controllers
go tcContrller.Run(5, ctx.Done())
...
select {}
```

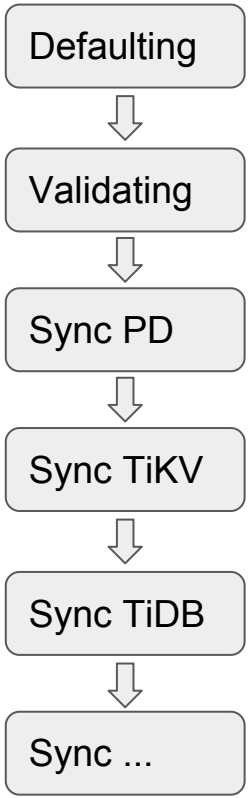
tidb-controller-manager - Watch Objects

```
func NewController(...) *Controller {
  ...
  // create a work queue
  c.queue = workqueue.NewNamedRateLimitingQueue(workqueue.DefaultControllerRateLimiter(), "tidbcluster")
  // add object to the queue on event
  informerFactory.Pingcap().V1alpha1().TidbClusters().Informer().AddEventHandler(cache.ResourceEventHandlerFuncs{
      AddFunc: c.enqueue,
      UpdateFunc: func(old, cur interface{}) {
          c.enqueue(cur)
      },
      DeleteFunc: c.enqueue,
  })
  ...
}

func (c *Controller) enqueue(tc *v1alpha1.TidbCluster) {
    key := getObjectKey(obj)
    c.queue.Add(key)
}
```

tidb-controller-manager - Start workers

```go
func (c *Controller) Run(workers int, stopCh <-chan struct{}) {
    defer c.queue.ShutDown()
    for i := 0; i < workers; i++ {
        go wait.Until(c.worker, time.Second, stopCh)
    }
    <-stopCh
}

func (c *Controller) worker() {
    for c.processNextWorkItem() {
    }
}

func (c *Controller) processNextWorkItem() bool {
    key, quit := c.queue.Get()
    if quit {
        return false
    }
    defer c.queue.Done(key)
    if err := c.sync(key.(string)); err != nil {
        c.queue.AddRateLimited(key)
    } else {
        c.queue.Forget(key)
    }
    return true
}
```

tidb-controller-manager - Reconcliation



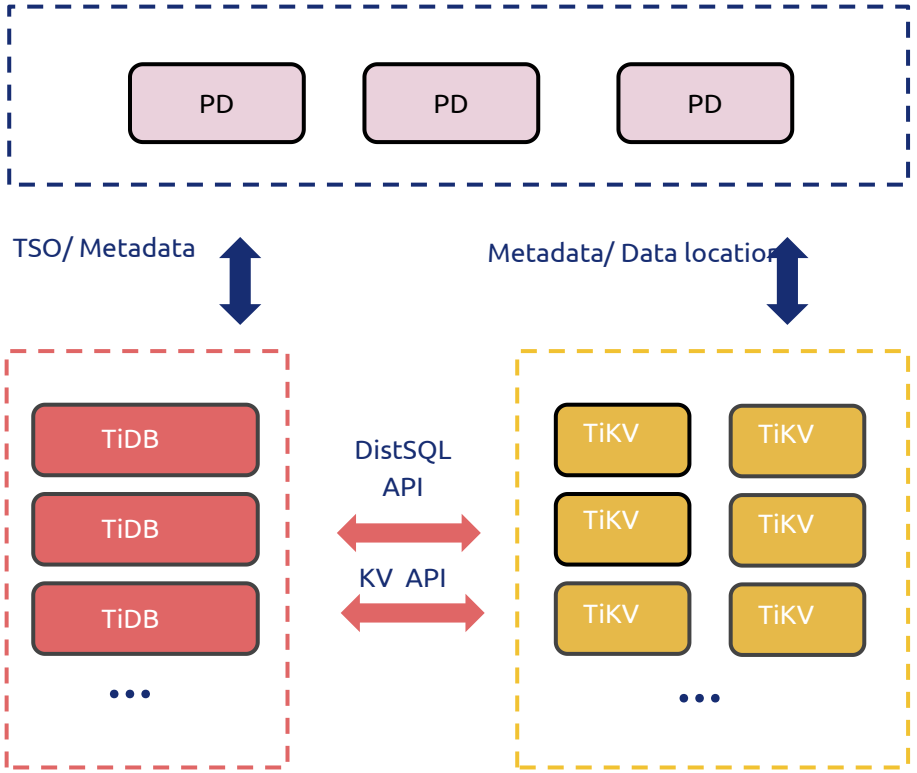

```go
func (c *Controller) Sync(key string) error {
  tc := getTidbClusterFromCache(key)
  c.defaulting(tc)
  if !c.validating(tc) {
    return nil
  }
  defer func() {
    c.updateStatus(tc)
  }()
  if err := c.syncPD(tc); err != nil {
    return err
  }
  if err := c.syncTiKV(tc); err != nil {
    return err
  }
  if err := c.syncTiDB(tc); err != nil {
    return err
  }
  ...
  return nil
}
```

Full Lifecycle management of a TiDB Cluster

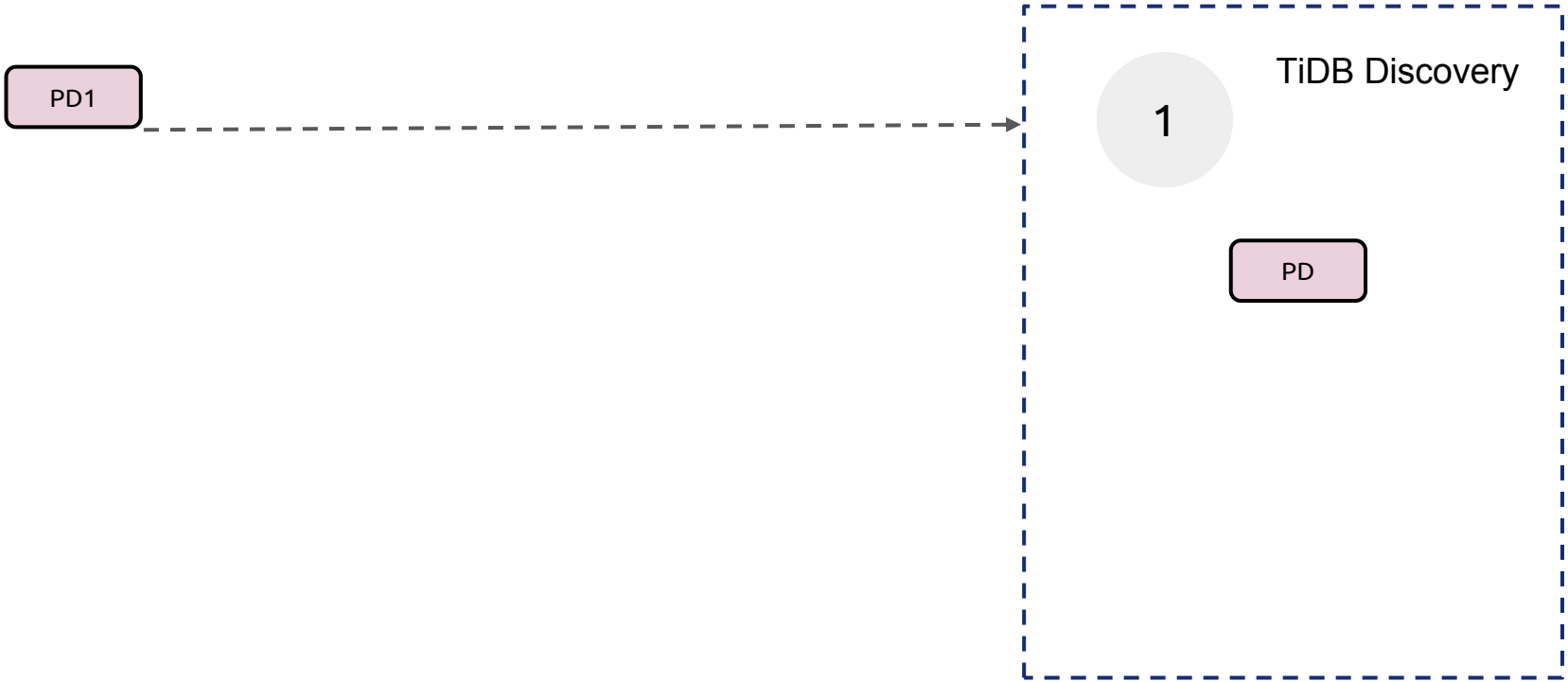
- Deploying
  - Bootstrapping
  - Configure services/configmaps, etc.
- Upgrading
  - Change version, config, etc.
- Scaling
- Automatic Failover
  - Create replacements for failed replicas

Deploying - Bootstrapping

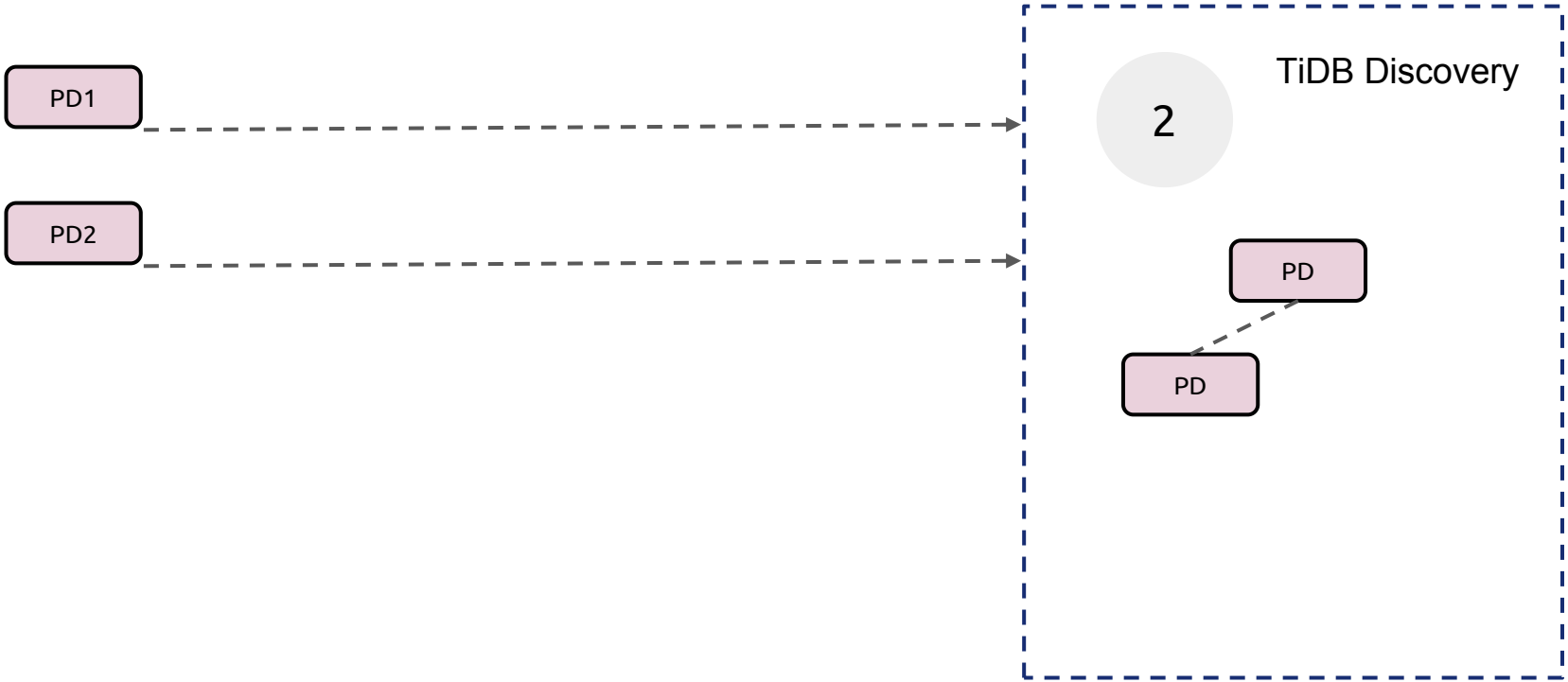
1. Bootstrap PD cluster with a discovery service
2. Start TiKV replicas and join the PD cluster
3. Start TiDB replicas and join the PD Cluster
4. Create TiDB Service...



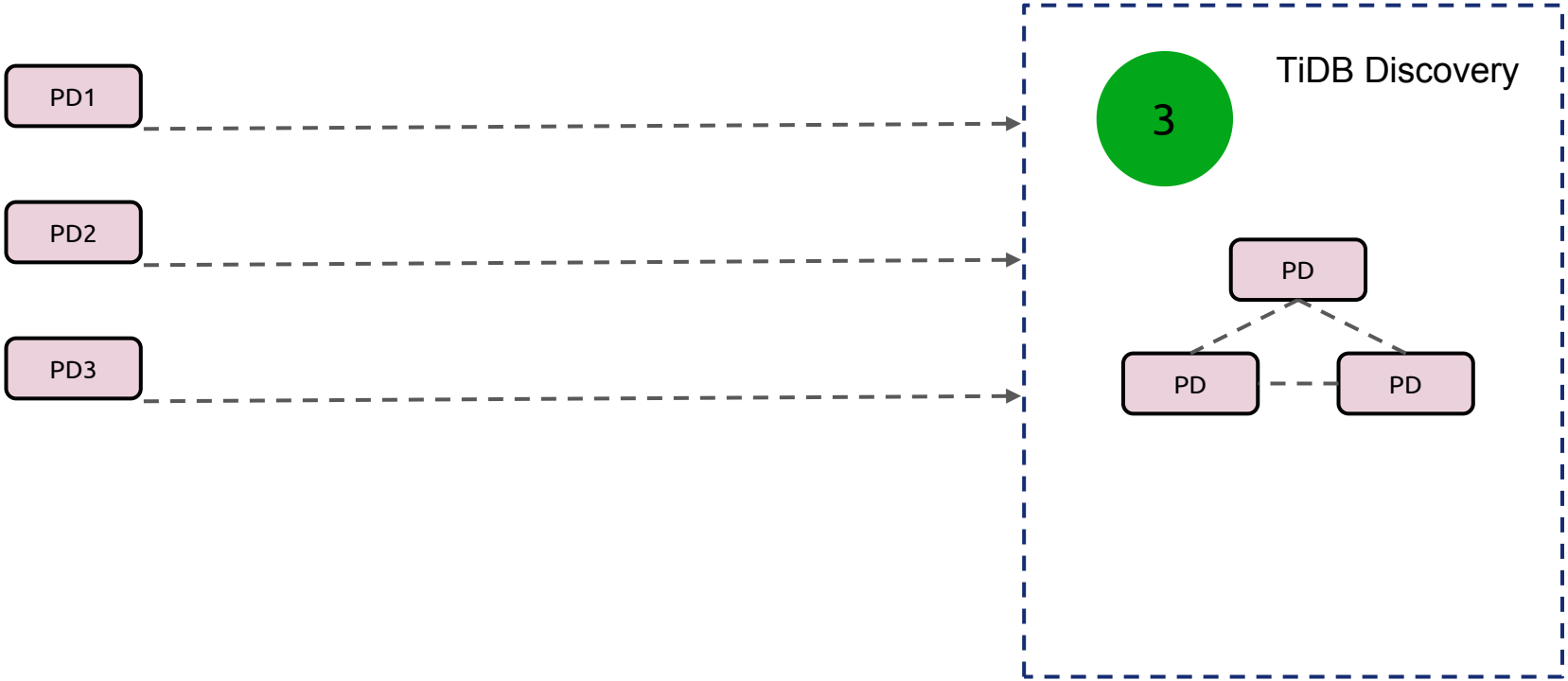
Deploying - Bootstrapping



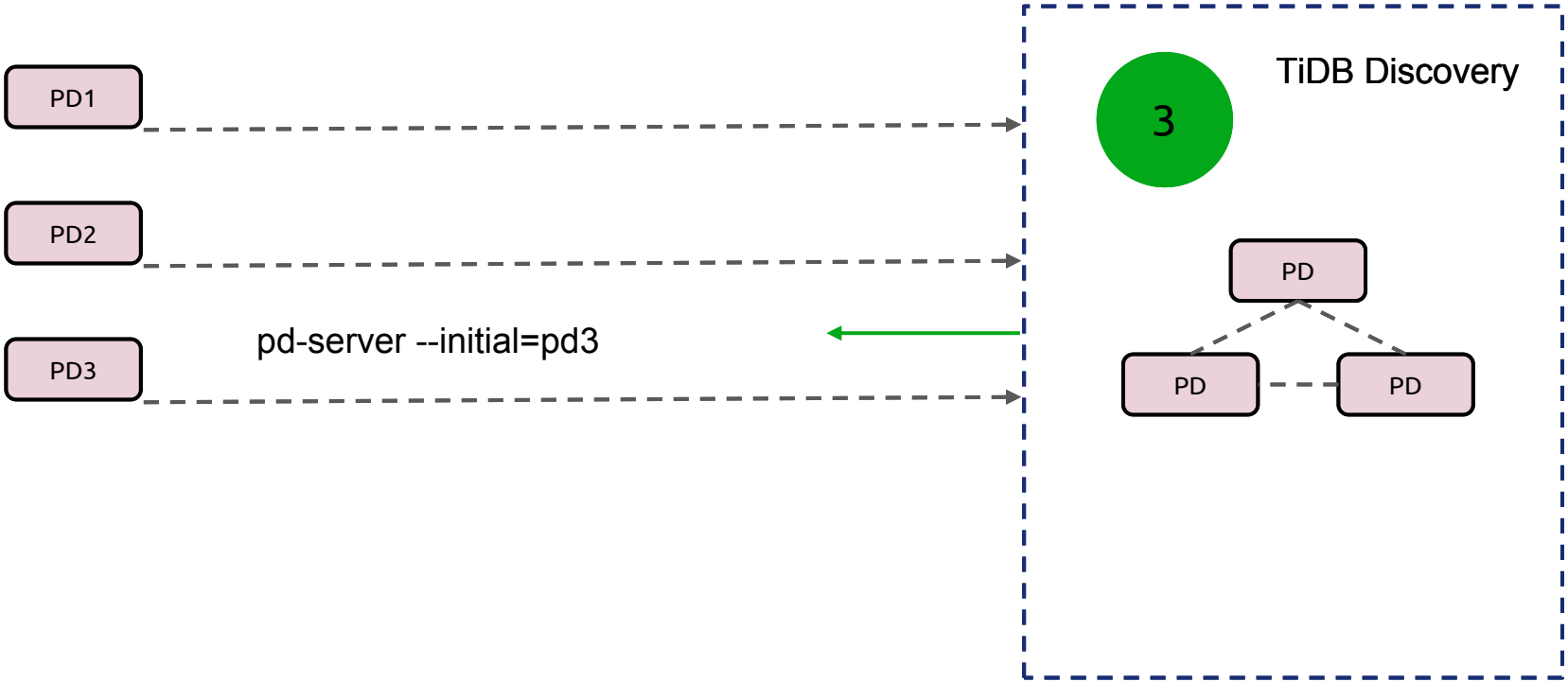
Deploying - Bootstrapping



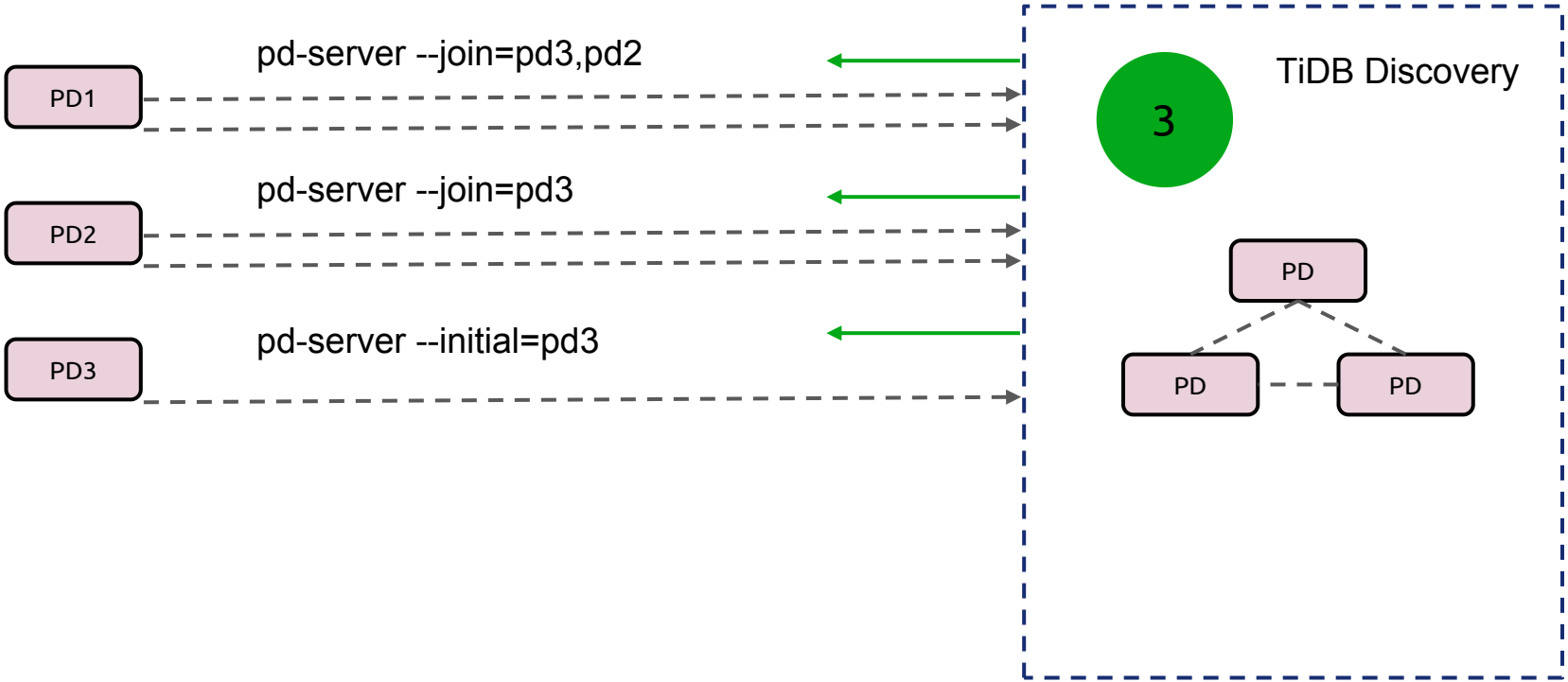
Deploying - Bootstrapping



Deploying - Bootstrapping



Deploying - Bootstrapping



Upgrading

It’s ok to upgrade a High Availability system like TiDB directly.

But, to avoid spikes we use StatefulSet partition to perform a phased roll out.

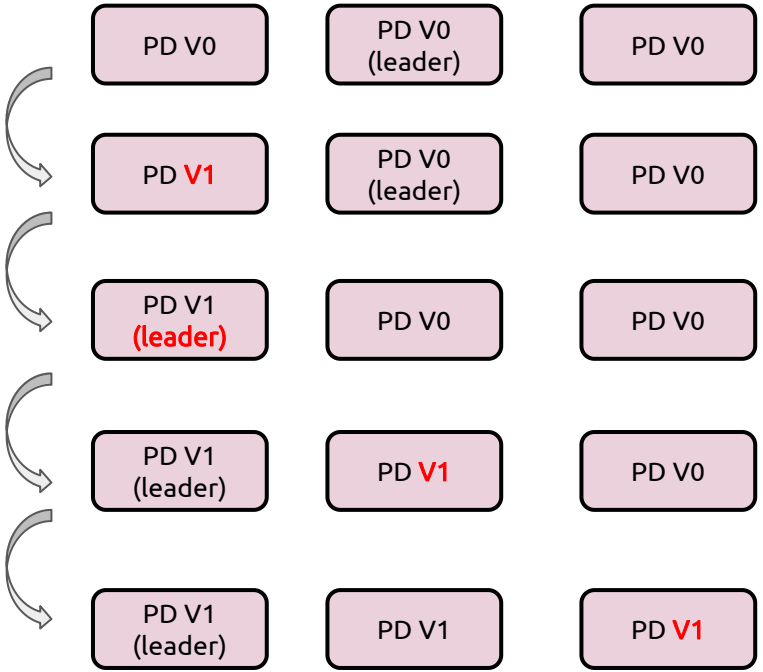
- Upgrade PD
- Upgrade TiKV
- Upgrade TiDB

```
status:
    currentRevision: <revision-v0>
    currentReplicas: 3
    updateRevision: <revision-v1>
    updateReplicas: 0

status:
    currentRevision: <revision-v0>
    currentReplicas: 2
    updateRevision: <revision-v1>
    updateReplicas: 1

status:
    currentRevision: <revision-v0>
    currentReplicas: 1
    updateRevision: <revision-v1>
    updateReplicas: 2

status:
    currentRevision: <revision-v1>
    currentReplicas: 3
    updateRevision: <revision-v1>
    updateReplicas: 3
```

Upgrading - PD

For each PD replica

- If it's the leader of PD cluster, transfer the leadership
- Upgrade



Upgrading - PD

For each PD replica

- If it's the leader of PD cluster, transfer the leadership
- Upgrade

```
for i in <pod ordinals from end to the start>:
    if isUpToDate(i):
        continue
    if not pd.IsLeader(i):
        setPartition(i)
    else
        pd.TransferLeader(i)
```

Upgrading - TiKV

For each TiKV replica

- Evict region leaders
- Wait for all region leaders are evicted or the timeout expired
- Upgrade

```
for i in <pod ordinals from end to the start>:
    if isUpToDate(i):
        continue
    if pd.RegionLeaderCount(i) == 0 or evictTimedOut(i):
        setPartition(i)
    else
        pd.evictRegionLeader(i)
```

Upgrading - TiDB

For each TiDB replica

- Upgrade

```
for i in <pod ordinals from end to the start>:
    if isUpToDate(i):
        continue
    setPartition(i)
```

Scaling In - PD

- Delete the member via PD API first
- After the member is deleted successfully, scale in the StatefulSet

```
1  for i in <pods need to delete>:
2      if pd.Delete(i):
3          delete(i)
4      else
5          return RequeueError
```

Scaling In - TiKV

- Delete the TiKV store via PD API first
- Wait for the store to be tombstoned
- Scale the statefulset

```
for i in <pods from the end to the start>:
    if pd.IsTombStoned(i):
        scaleIn(i)
    else
        pd.DeleteStore(i)
```

Scaling In - TiDB

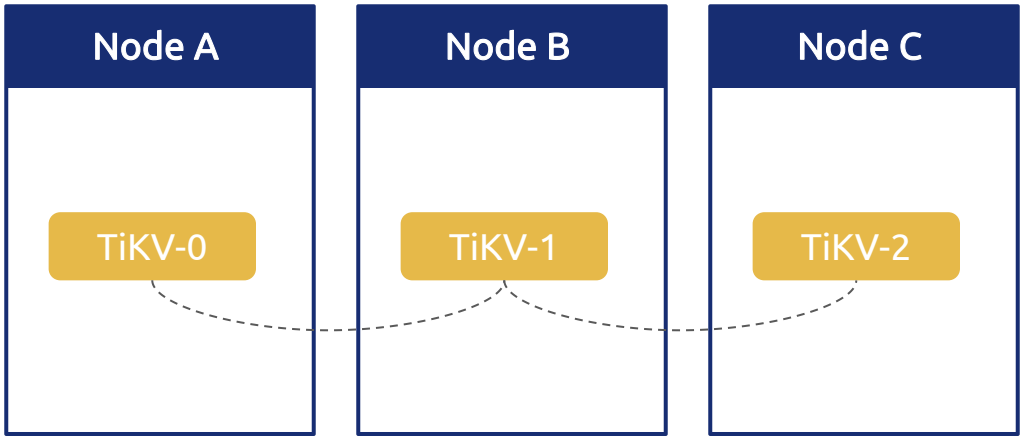
- Scale the statefulset directly

Scaling Out

- Clean retained PVCs if necessary
- Scale the statefulset
- Wait for new replicas to join the cluster

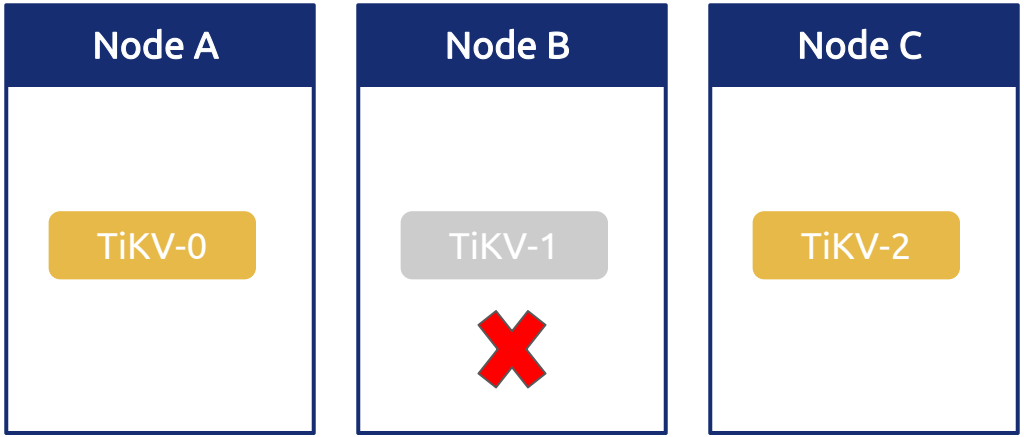
Automatic Failover

- 3 default replicas in each raft group can tolerate one member failure

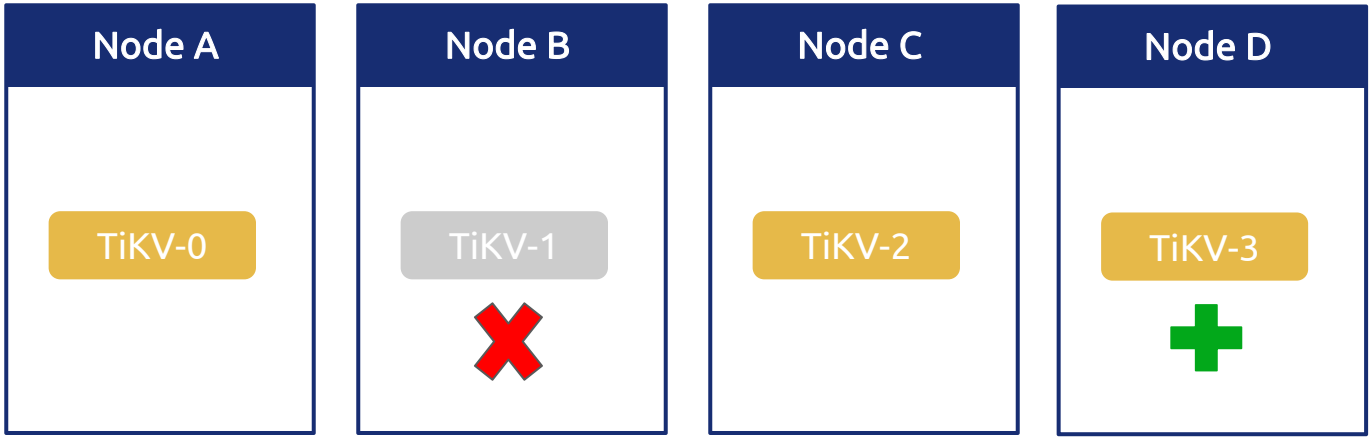


Automatic Failover

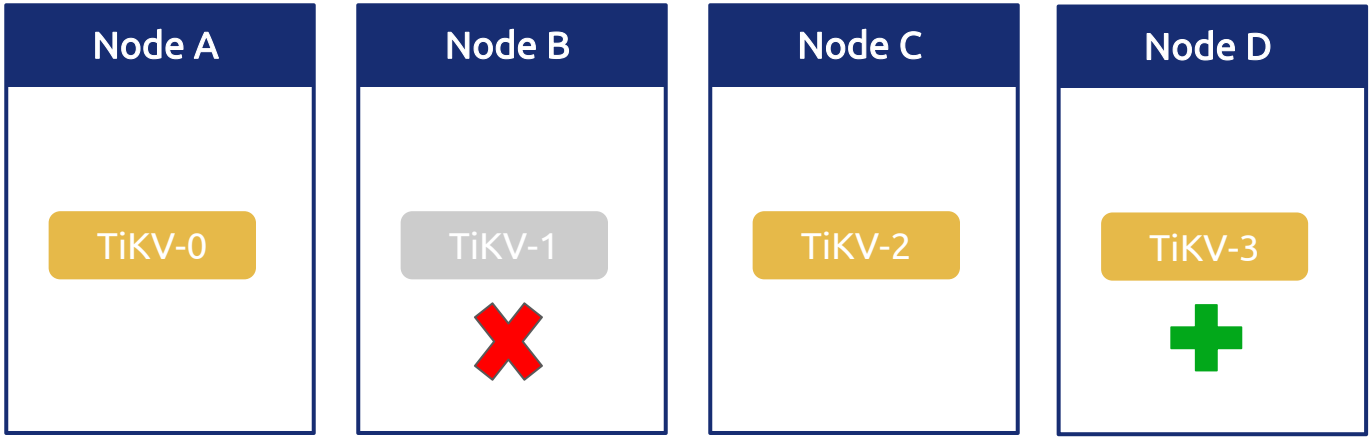
- What if a repliac is down for a long time
  - It’s dangerous!
- What can we do?



Automatic Failover



Automatic Failover



Automatic Failover

```
1   for store in <all tikv stores>:
2       if pd.IsDown(store) and sinceDownTime(store) > 5m:
3           markStoreFailed(store)
4   scaleStatefulSet(currentReplicas + failedStores)
```

- A store is marked **Down** by PD when it's unreachable for **30m** (by default)
- A store is marked **Failed** by TiDB Operator when it's **Down** for **5m** (by default)
- The controller will **increase the replicas** by the number of failed replicas

Summary

- TiDB Operator 简介
  - Cloud Native Era: Full Automation
  - Kubernetes Pattern: Declarative Model
  - Operator Pattern: Extending Kubernetes
- 扩展 Kubernetes 的几种方式
  - Custom Resource
  - Custom Controller
  - Scheduler Extender
  - Admission Webhook
- TiDB Operator 实现
  - the tidb-controller-manager
  - Full lifecycle management of a TiDB Cluster
    - Deploying, Upgrading, Scaling, Automatic Failover, etc.

Thank You!

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

Presented by Yecheng Fu (@cofyc)