Zero-downtime K8s migration of 14K Apache Pinot fleet at LinkedIn

Myeongjae “Tony” Song

March 2026

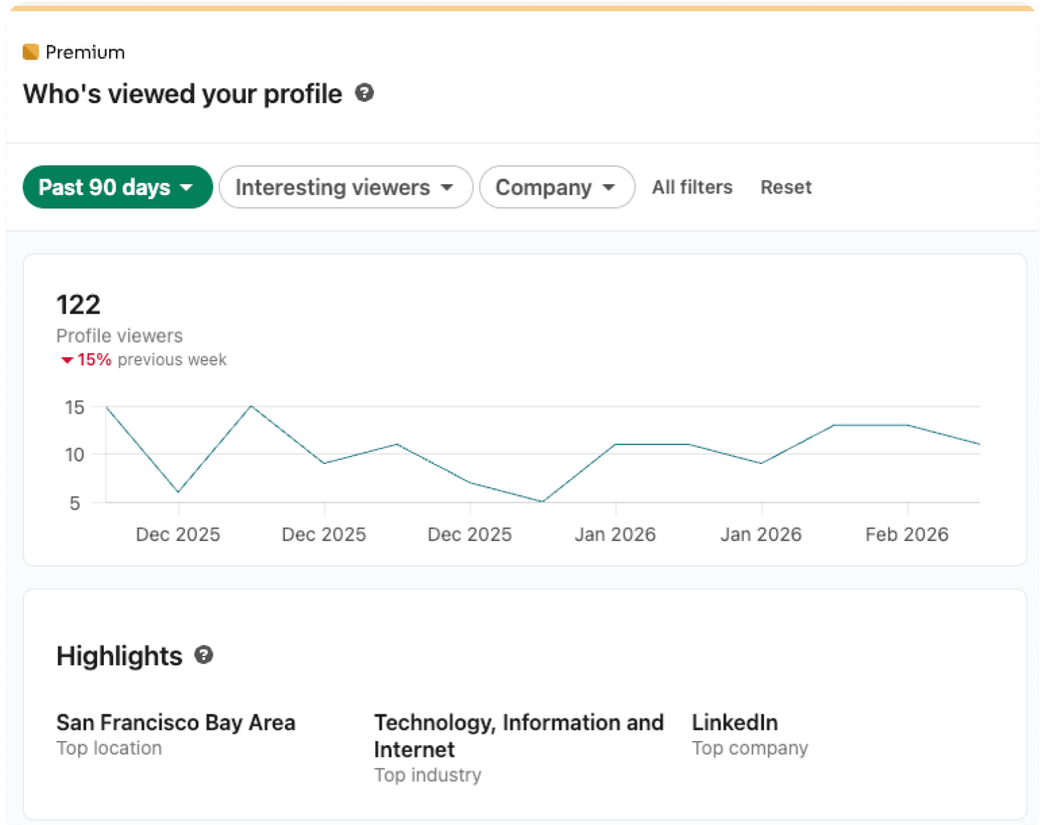
Agenda

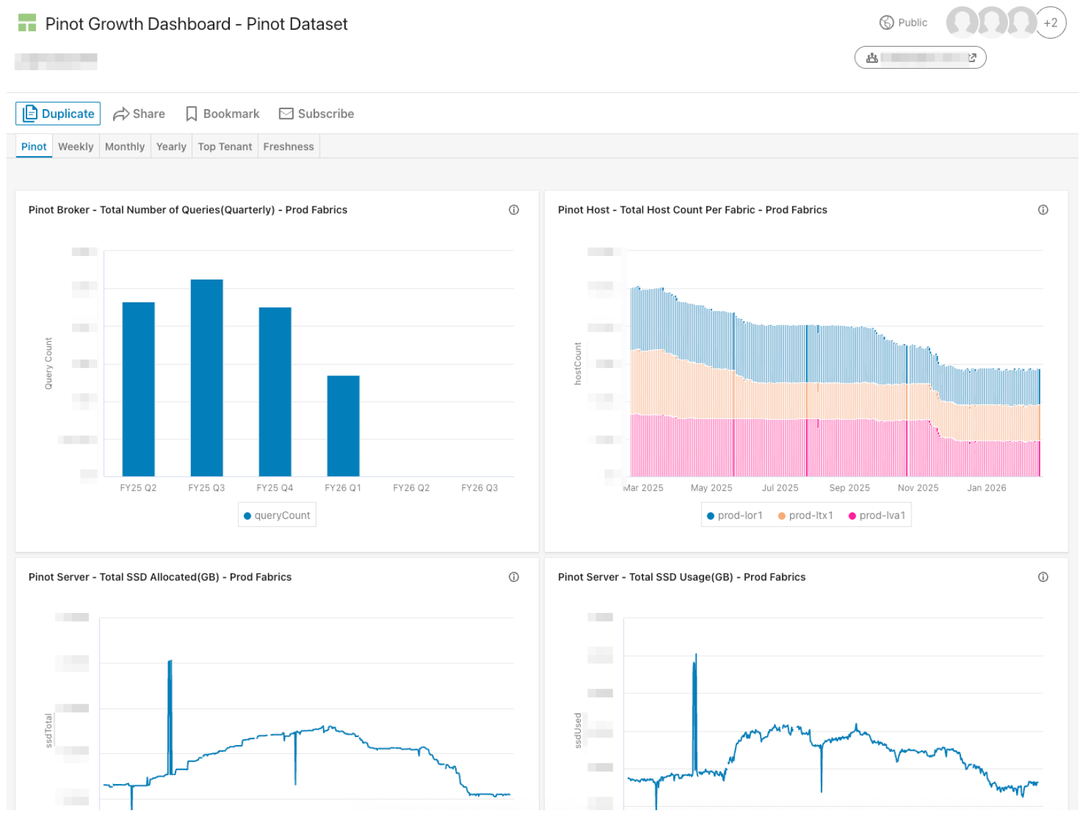
- What is Apache Pinot?
- Background
- Prereq: Availability Zone-Aware Shard Placement
- Prereq: Containerization
- Migration Orchestrator: Design
- Migration Orchestrator: Execution

Real-time distributed OLAP datastore built for low-latency, high-throughput analytics.

Apache Pinot @LinkedIn

Powers low latency online analytics for both site-facing and internal products





Member Facing & Enterprise Analytics

Internal Products & BI

Pinot Architecture

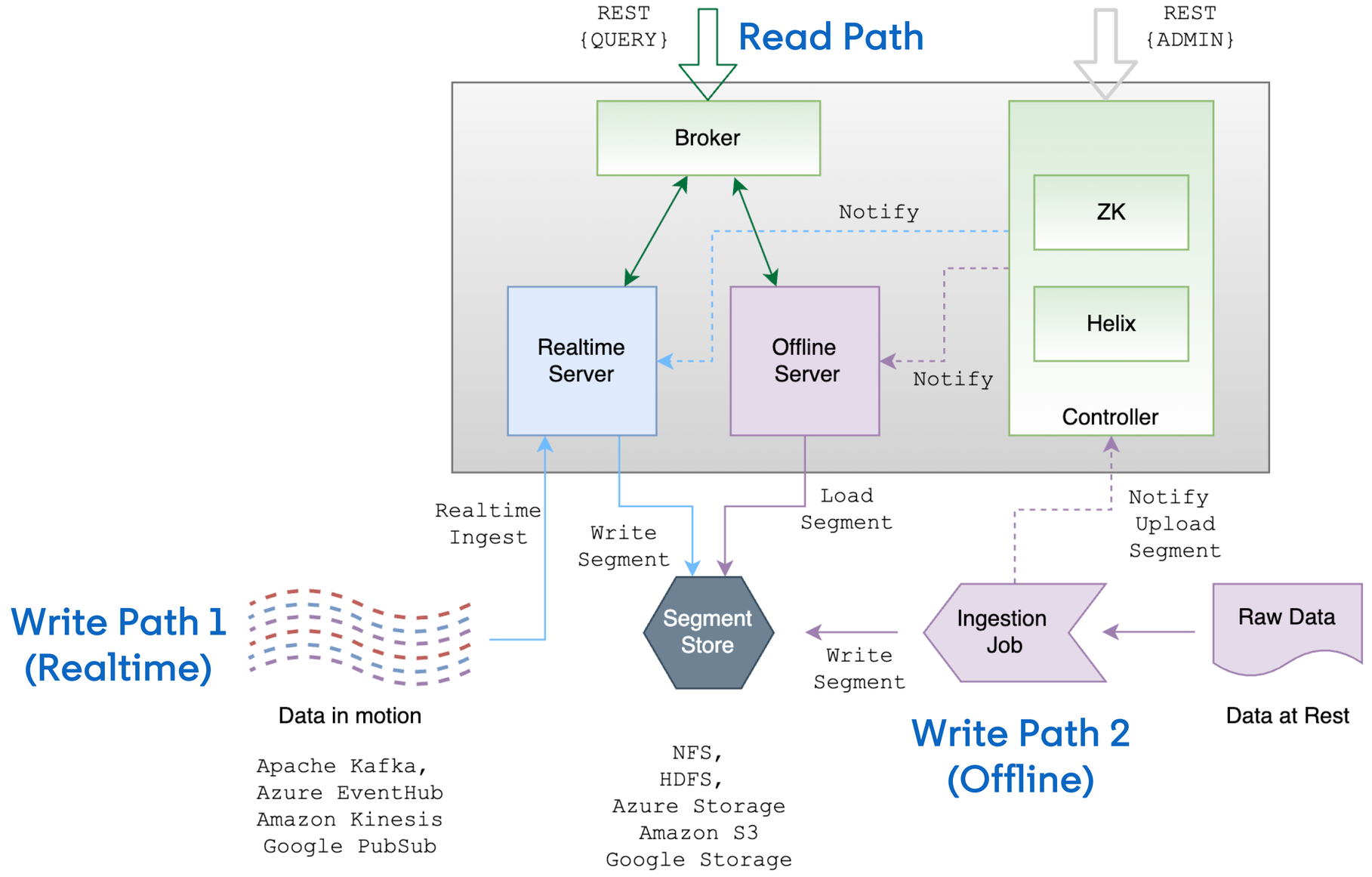


Image source: Apache Pinot doc

Apache Pinot @LinkedIn - Scale

300K+	10,000+	2,000+	4,500+
Queries Per Second	Serving Hosts	Broker Hosts	Pinot Tables

How can I migrate 14K stateful hosts?

Can’t you turn off linkedin.com for two weeks for migration?

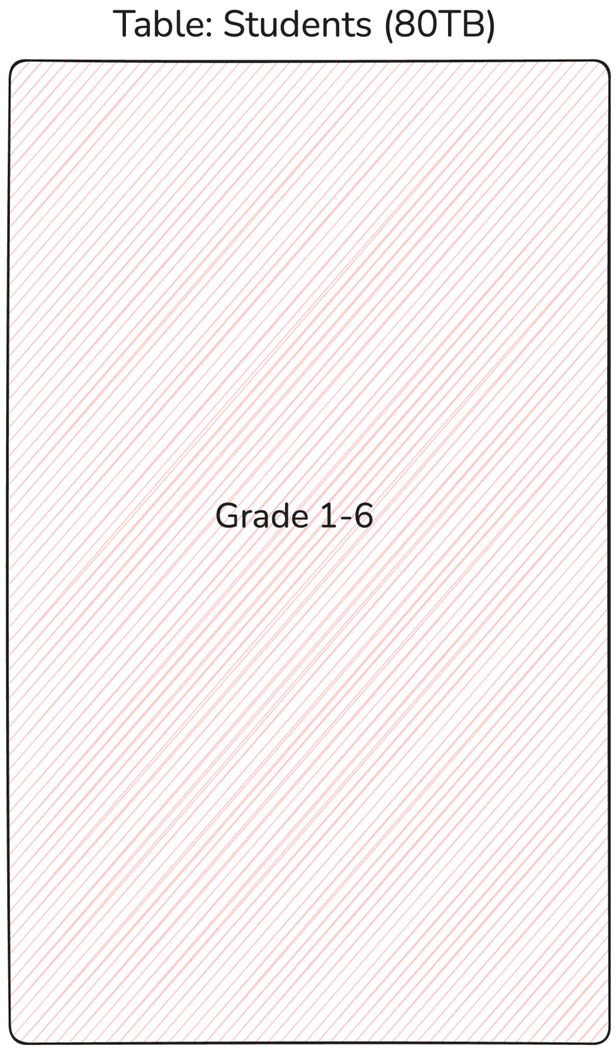
A very reasonable person may say

How can I migrate 14K stateful hosts with no-downtime and no query latency impact?

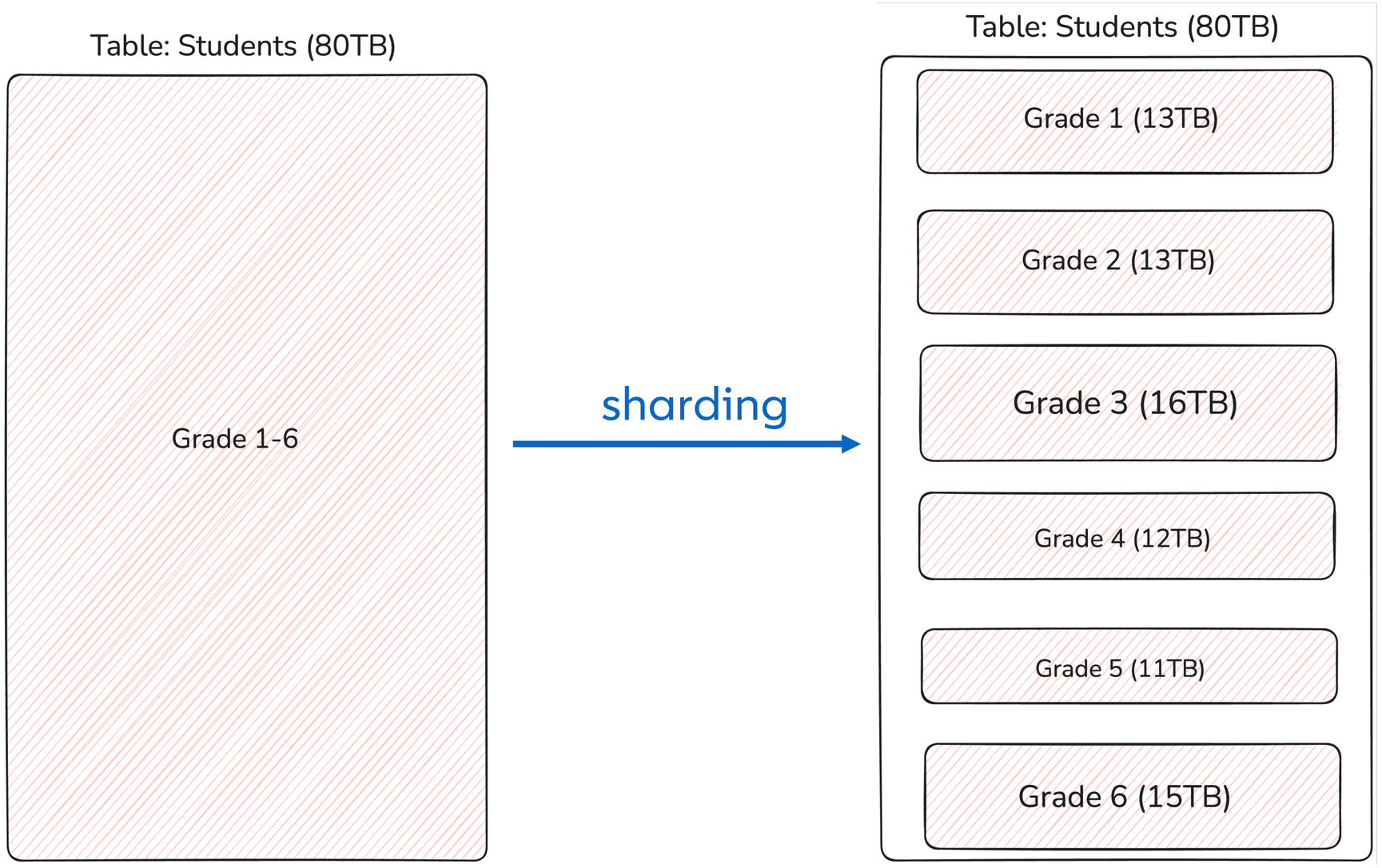
Background

2

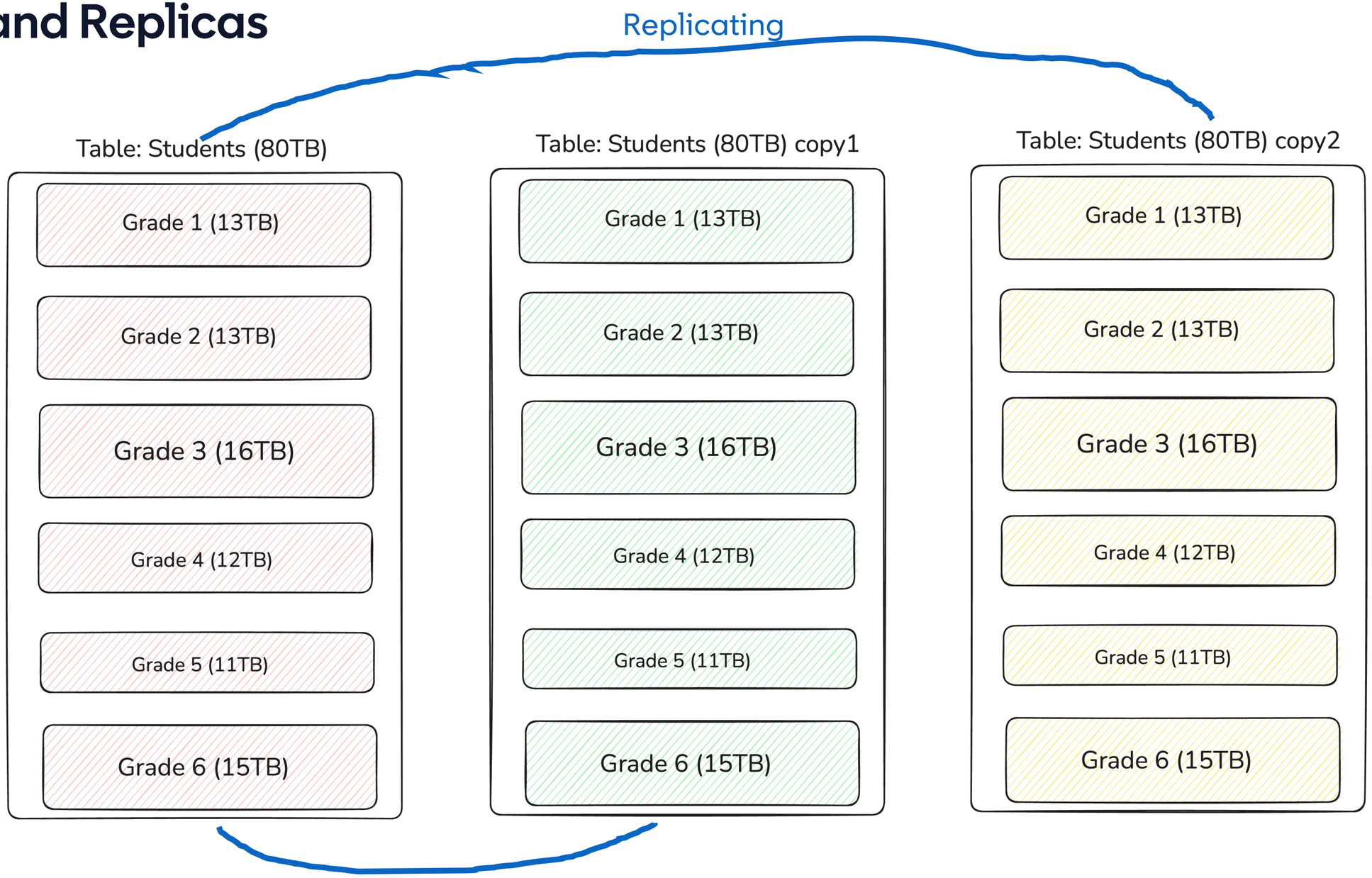
Shards and Replicas



Shards and Replicas



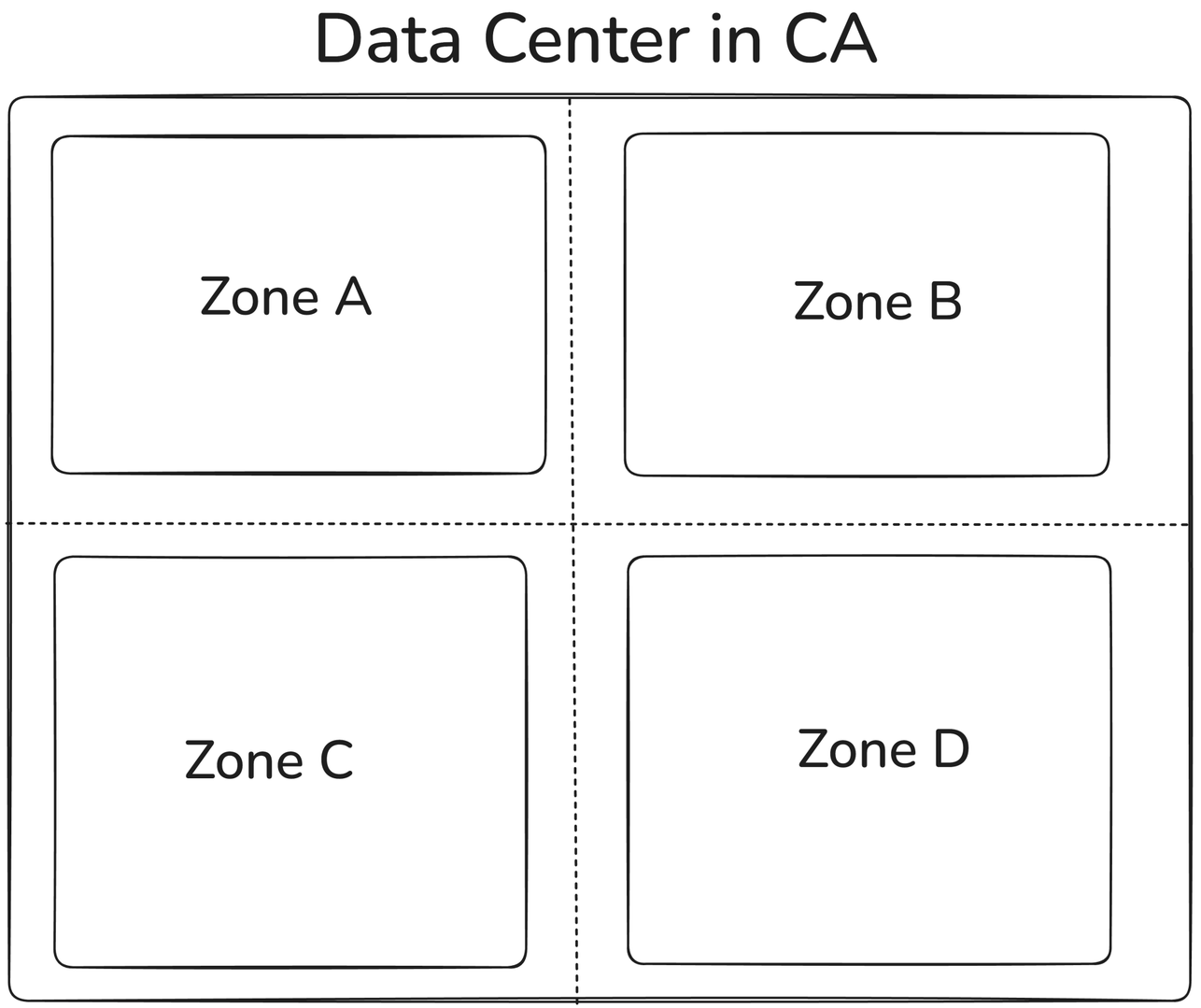
Shards and Replicas



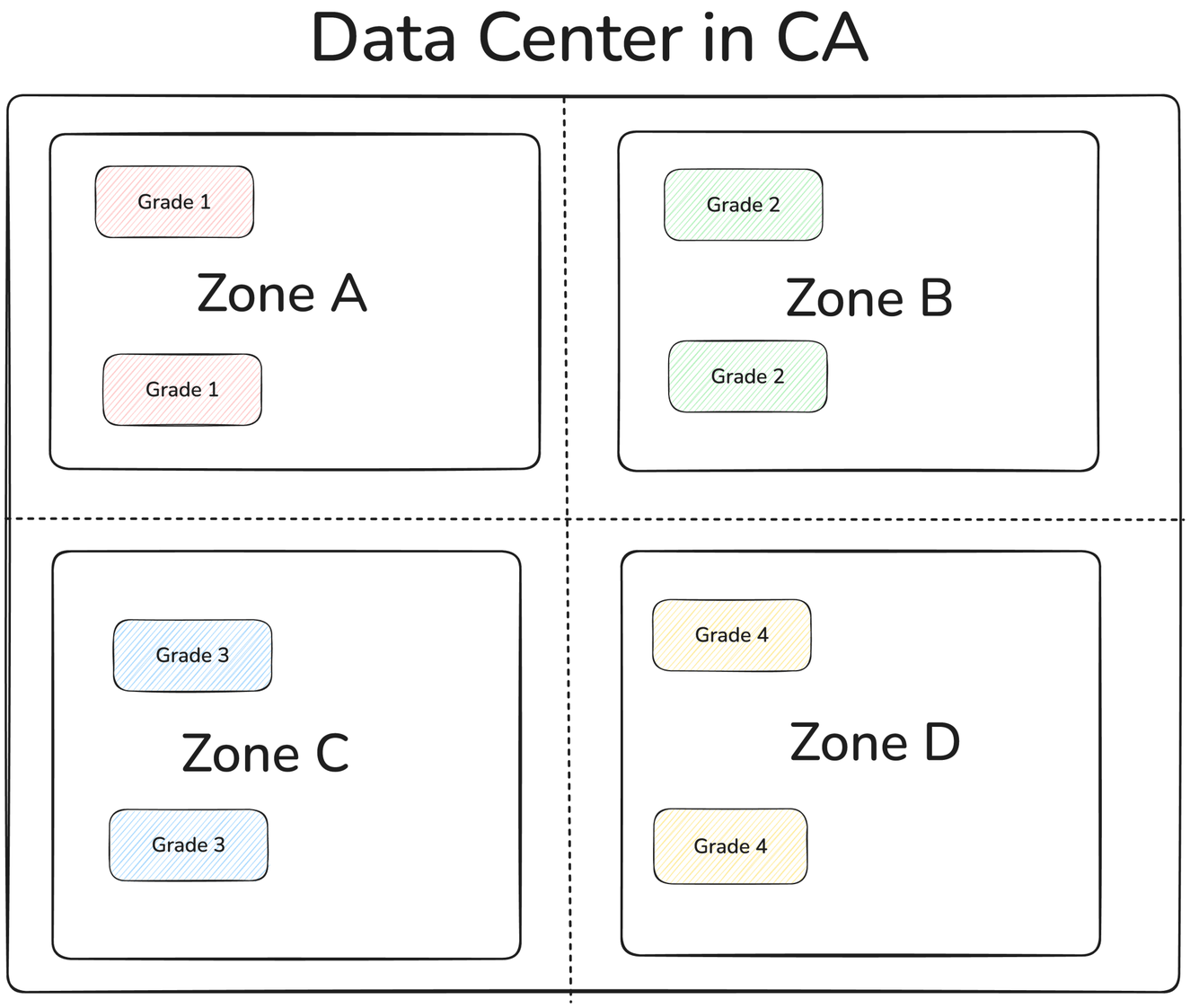
Prereq: Availability Zone-Aware Shard Placement

3

Availability Zones

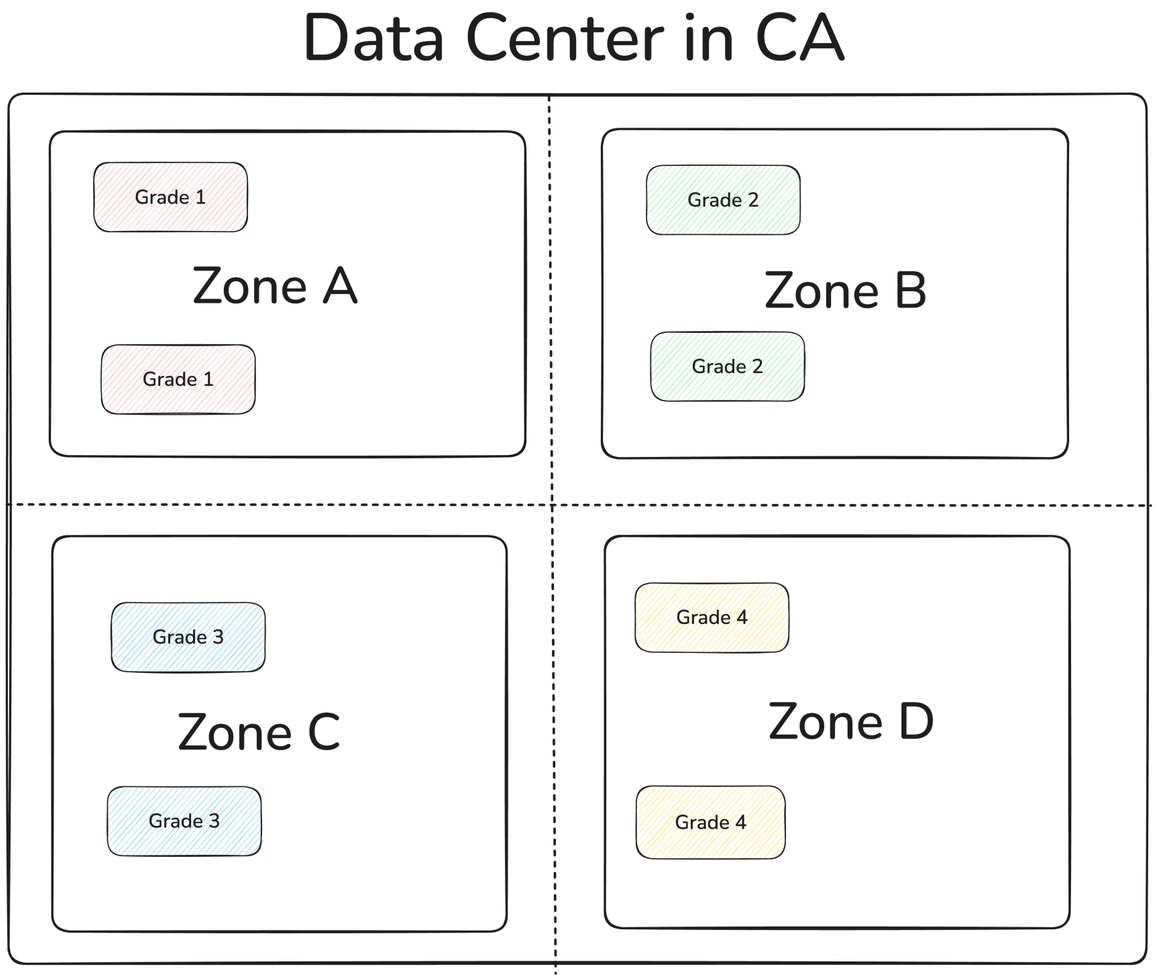


Shard Placement in Availability Zones

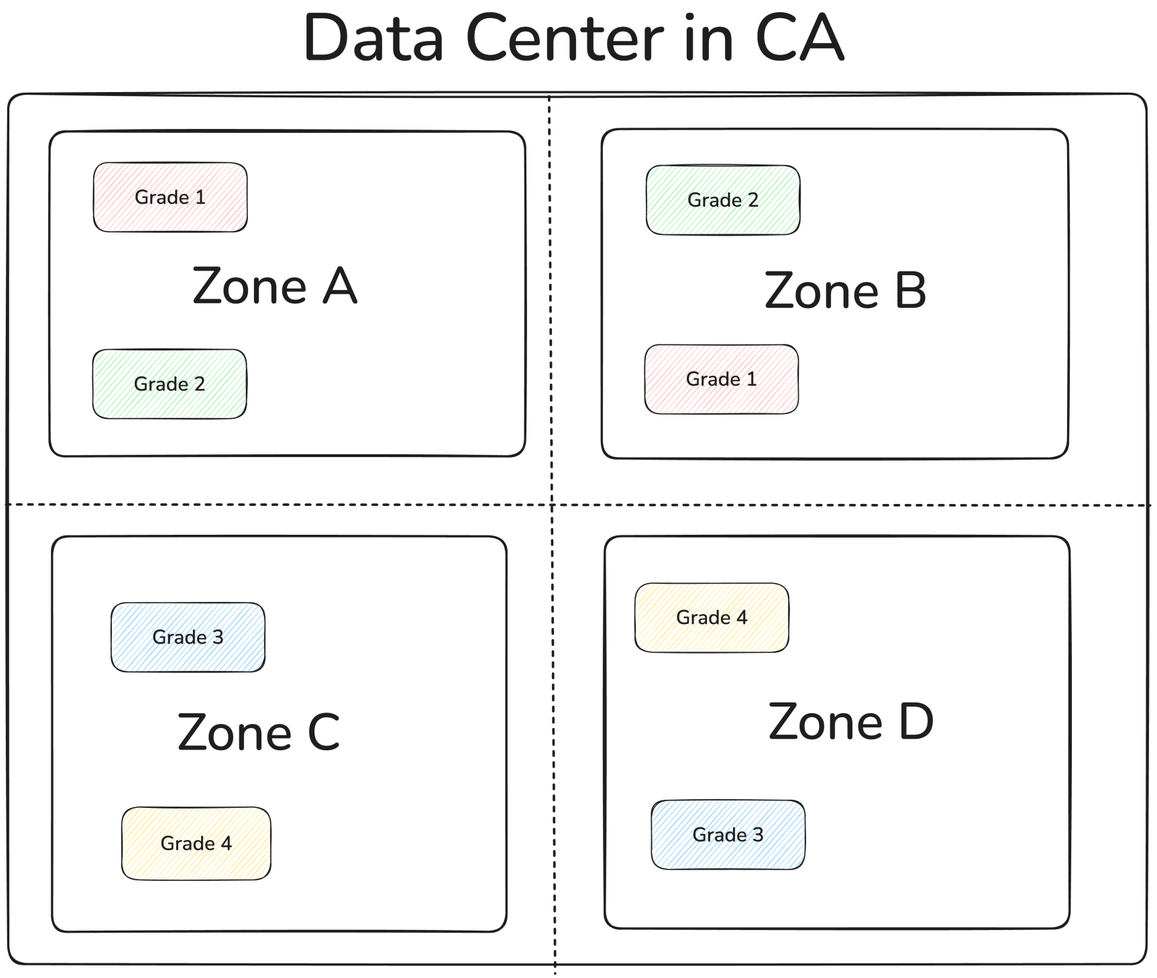


4 grades
With 2 replicas each

Shard Placement in Availability Zones



Zone non-diverse shard placement

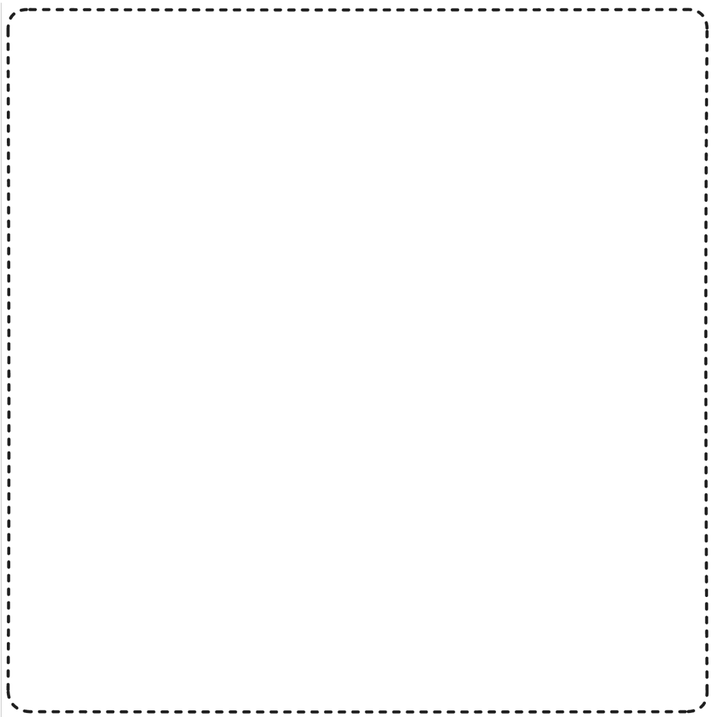


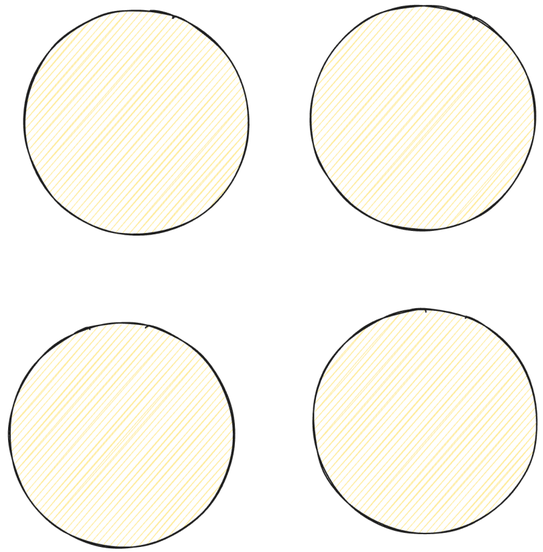
Zone diverse shard placement

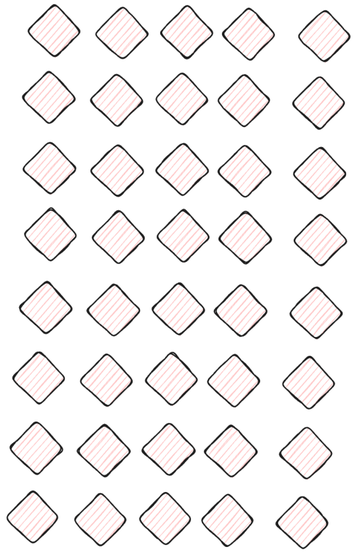
Shards: 10K+
Replicas: 40+
Availability Zones: 10+

...for a big Pinot table

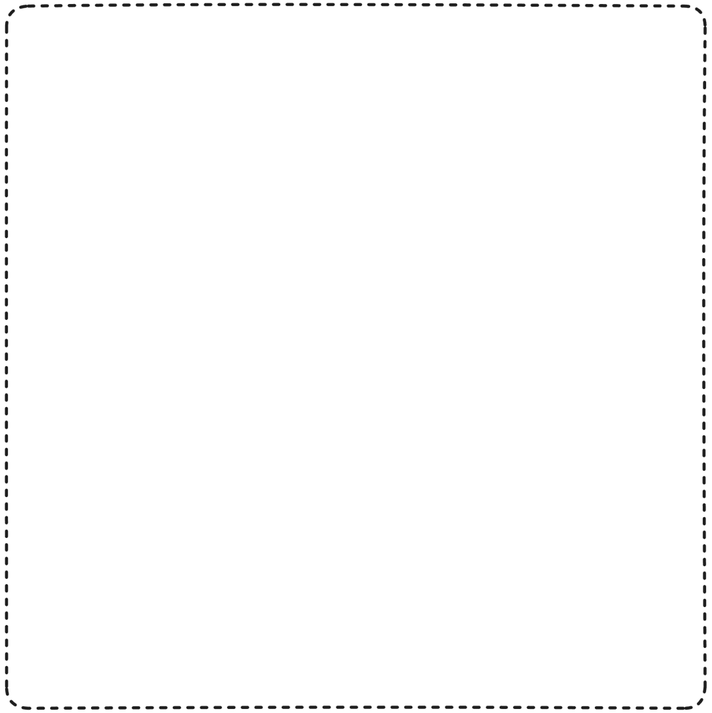
Adjustable Knobs for Shard Placement

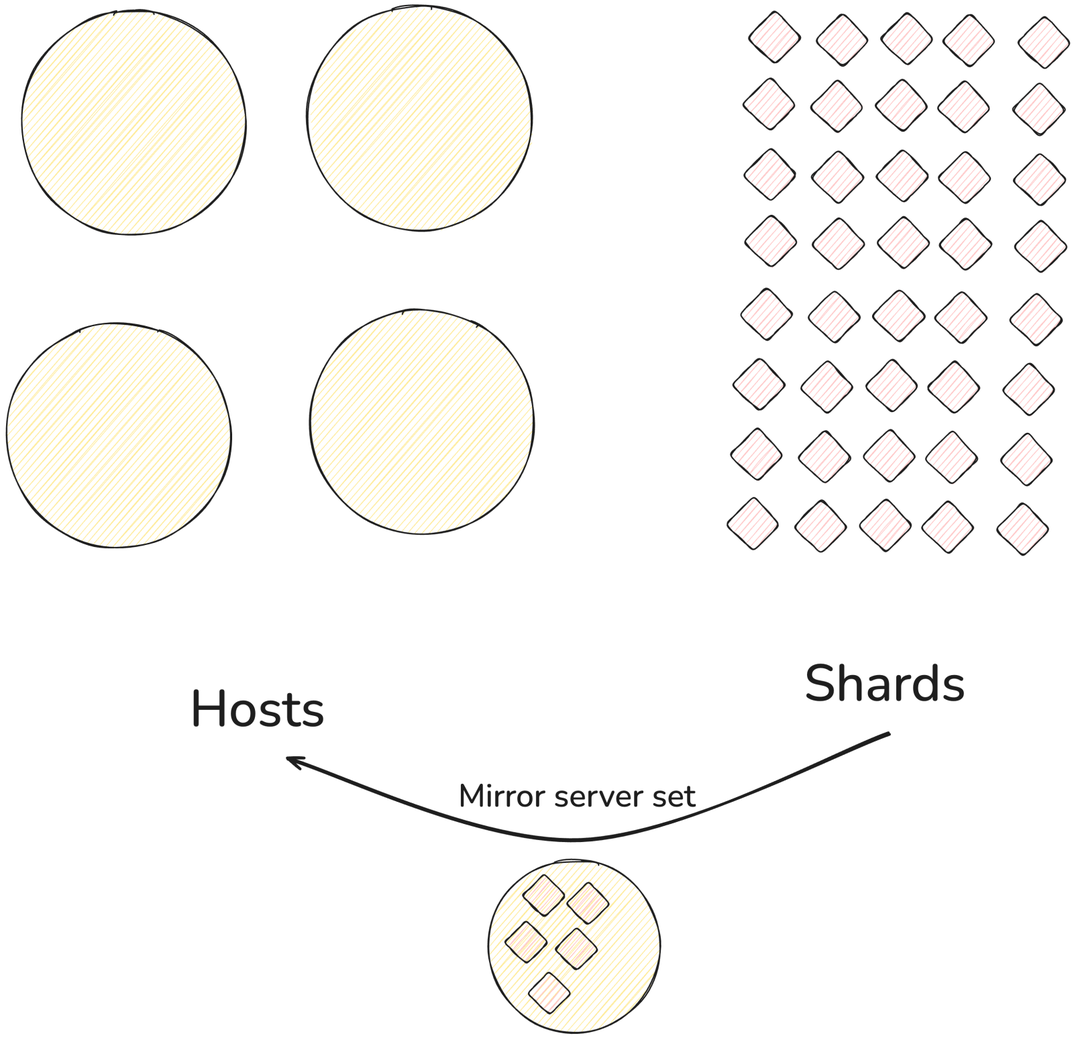




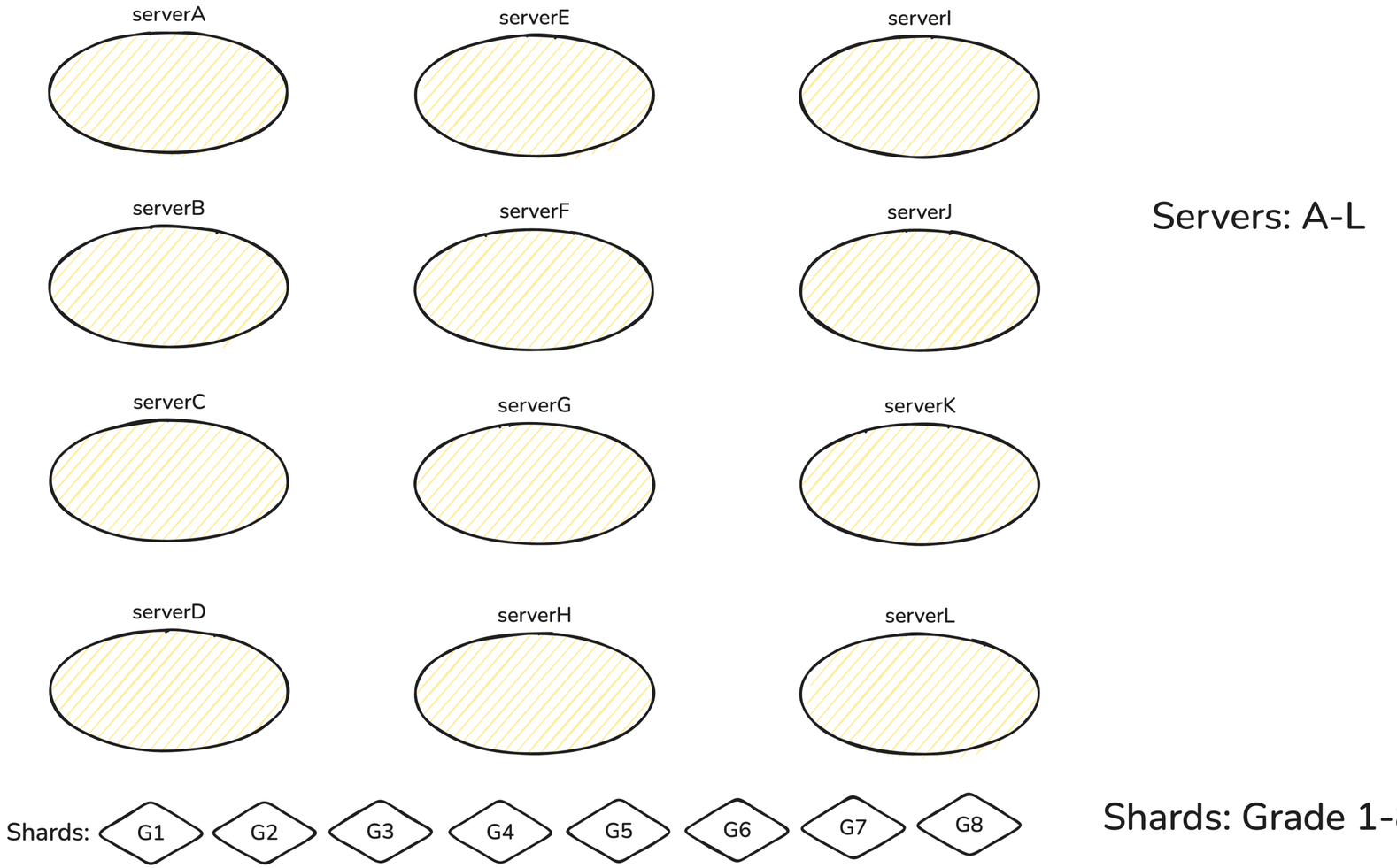


Adjustable Knobs for Shard Placement





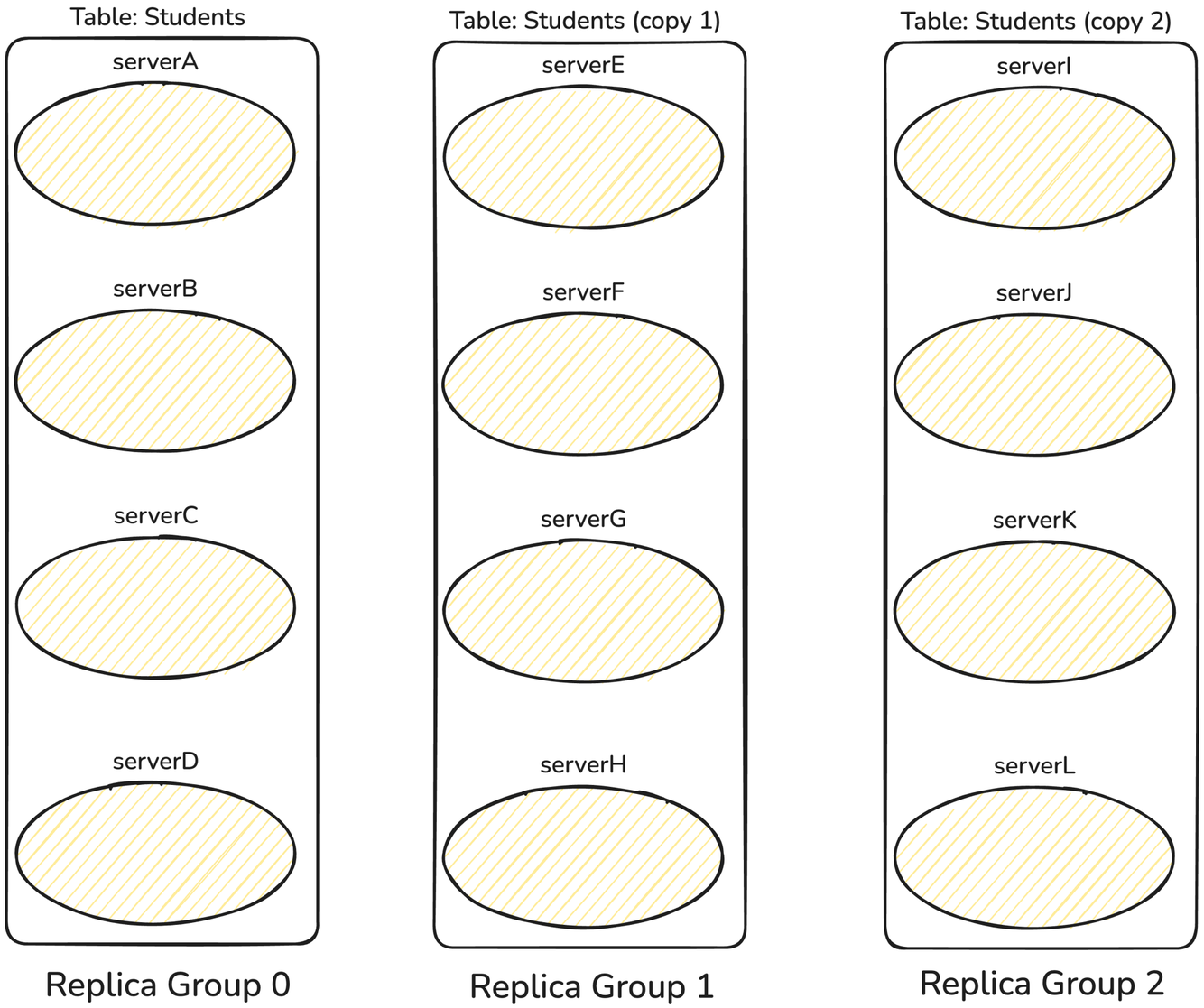
Replica Groups: How to place shards in hosts



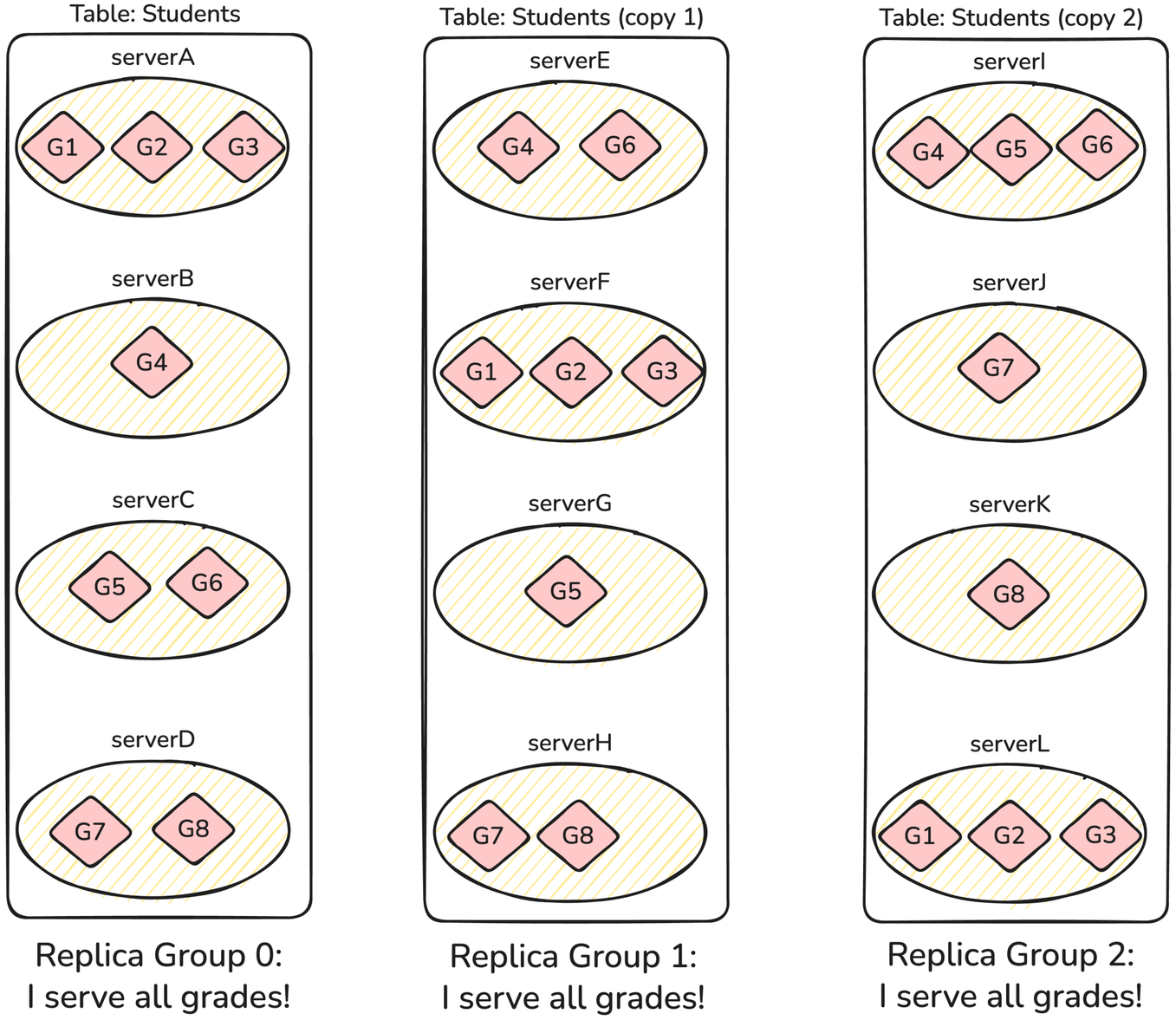
Servers: A-L

Shards: Grade 1-8

Replica Groups: How to place shards in hosts



Replica Groups: How to place shards in hosts

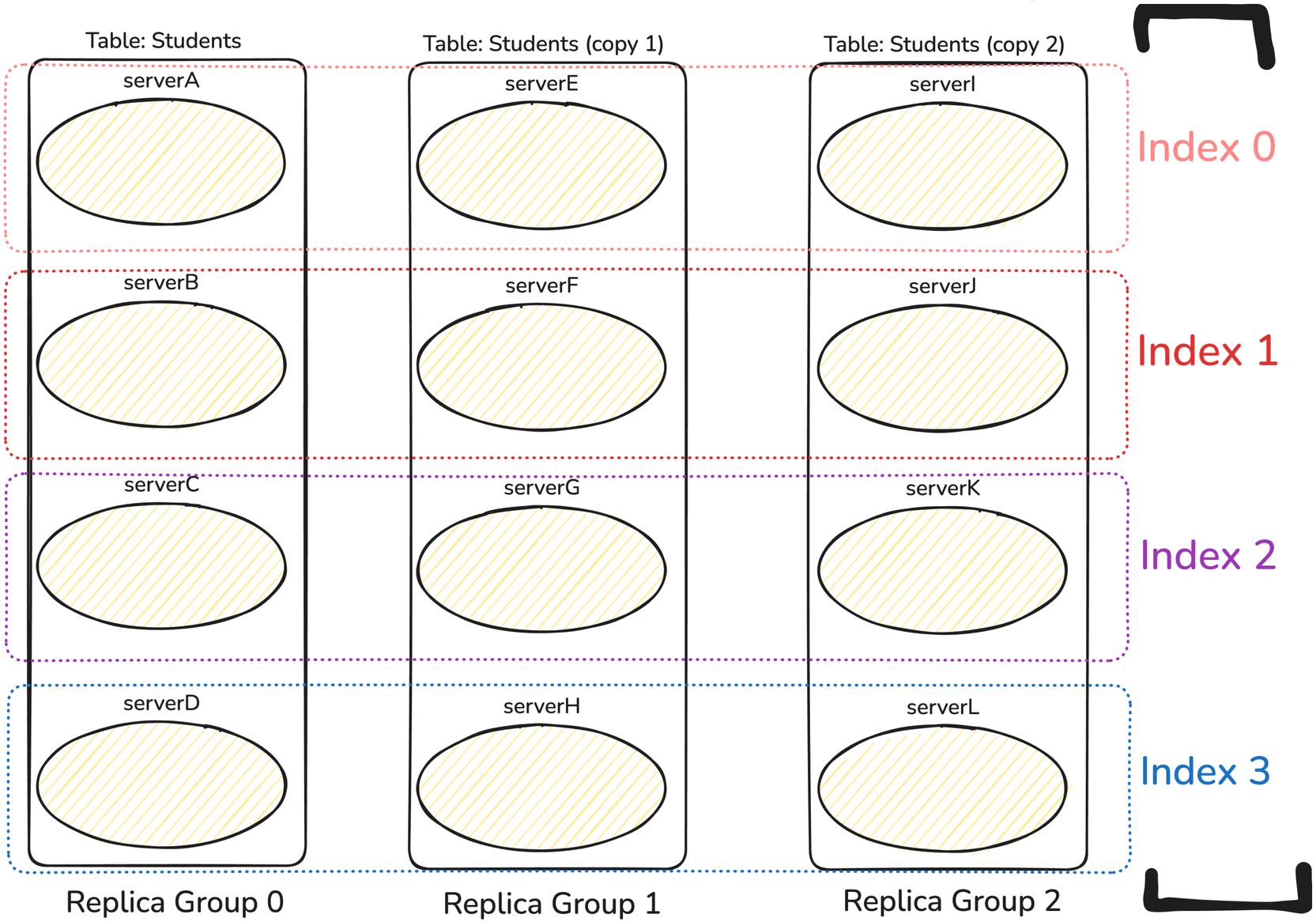


Replica Group 0:
I serve all grades!

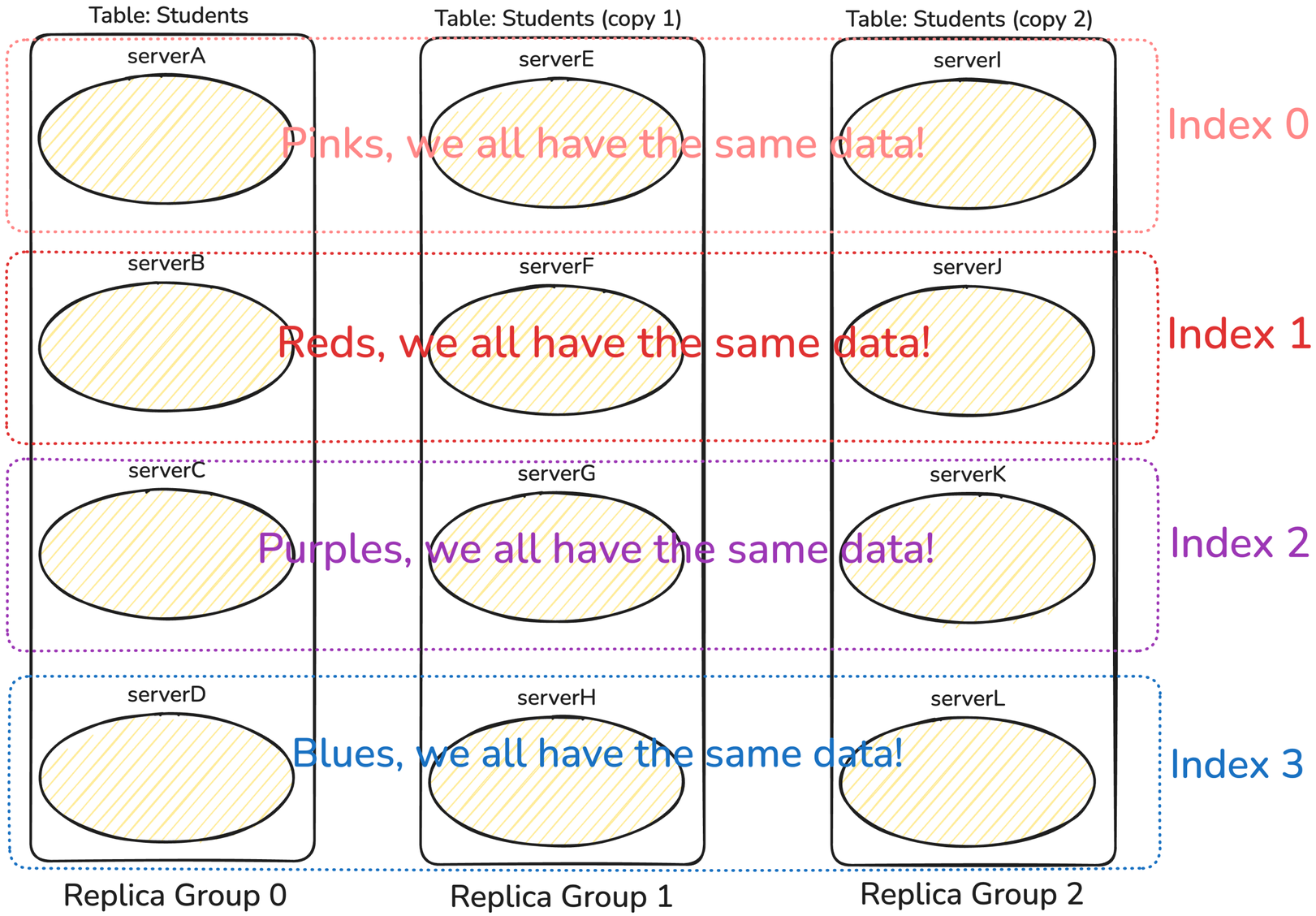
Replica Group 1:
I serve all grades!

Replica Group 2:
I serve all grades!

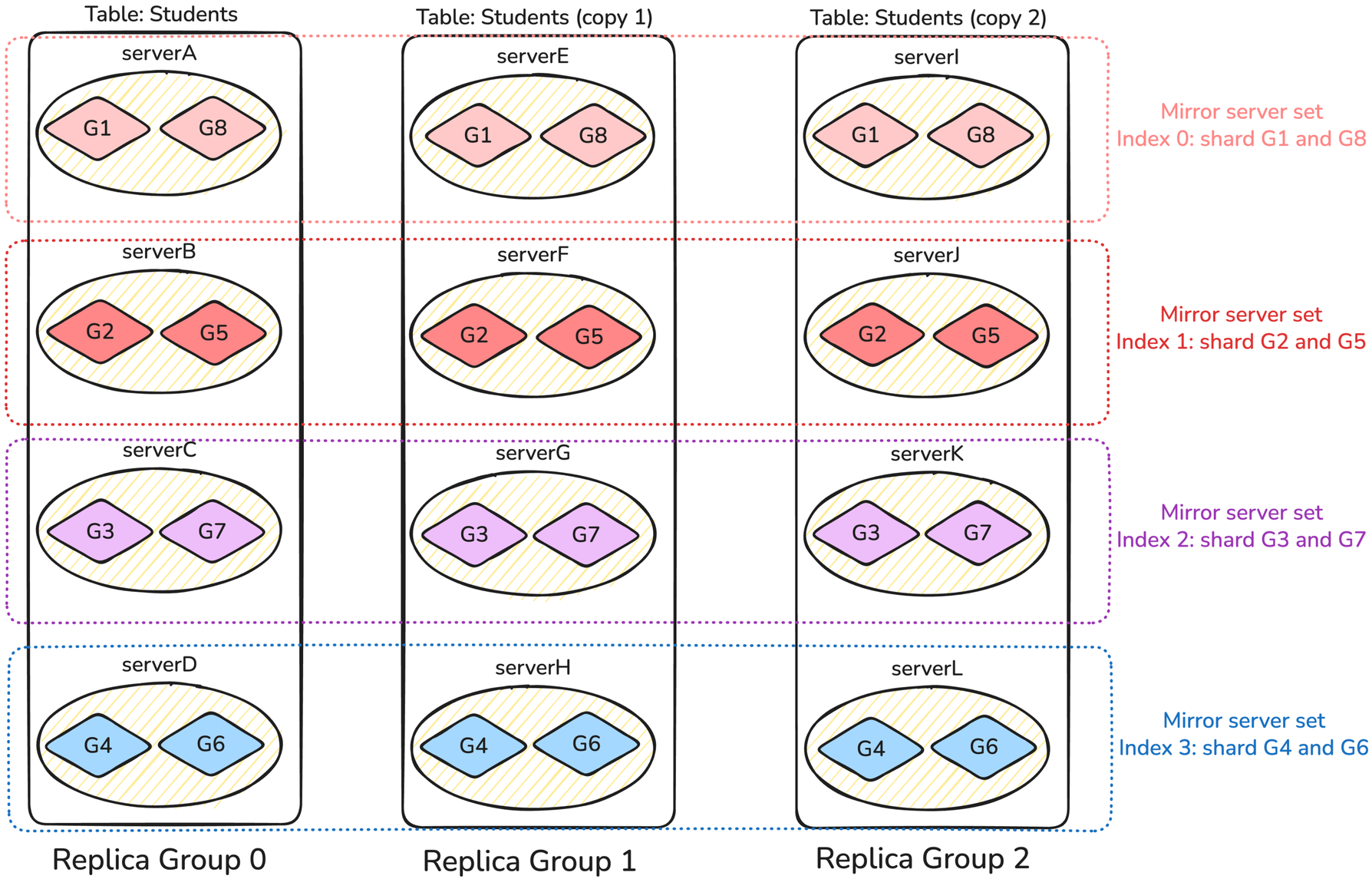
Mirror Server Set: How to place shards in hosts predictably



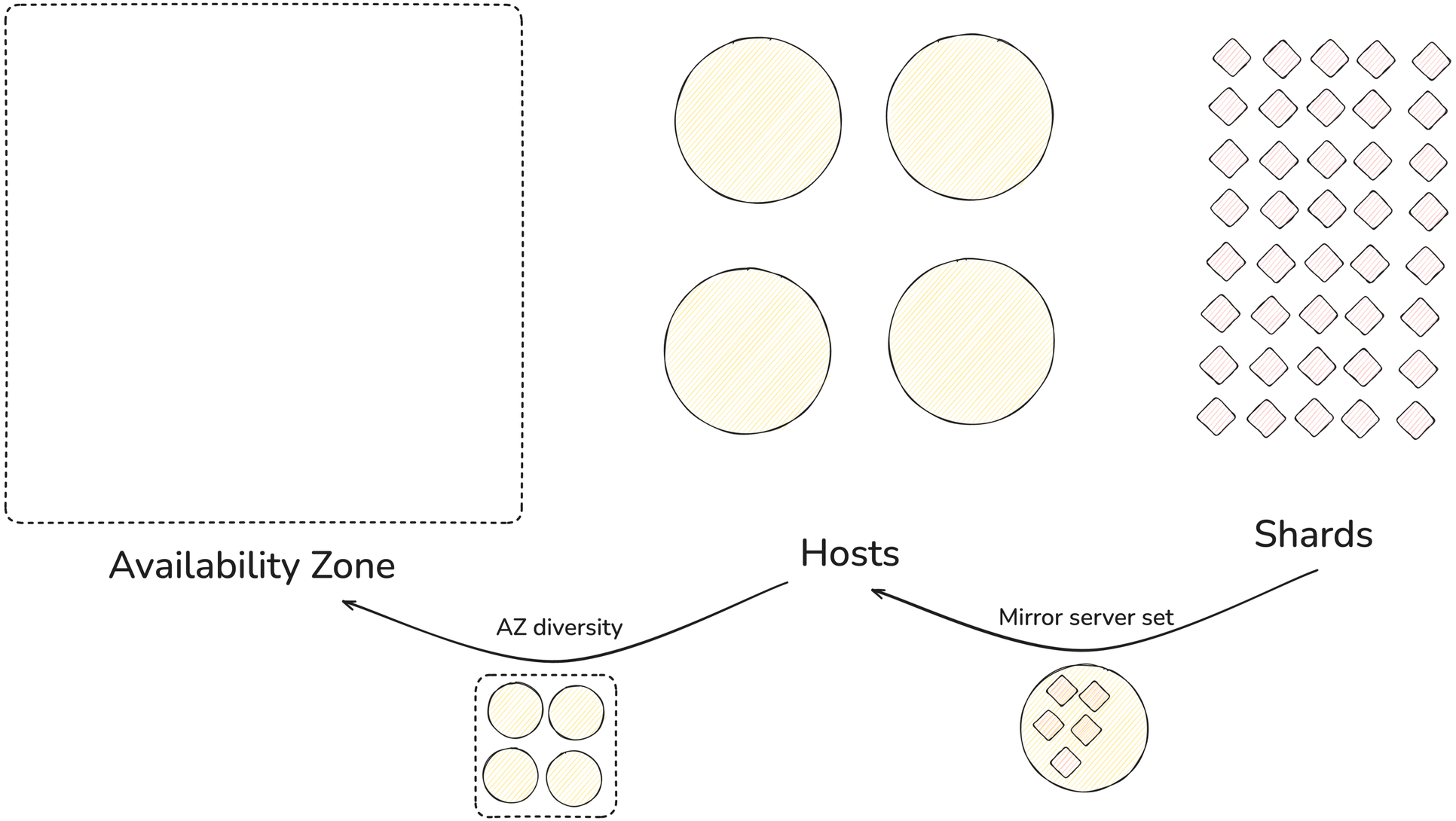
Mirror Server Set: How to place shards in hosts predictably



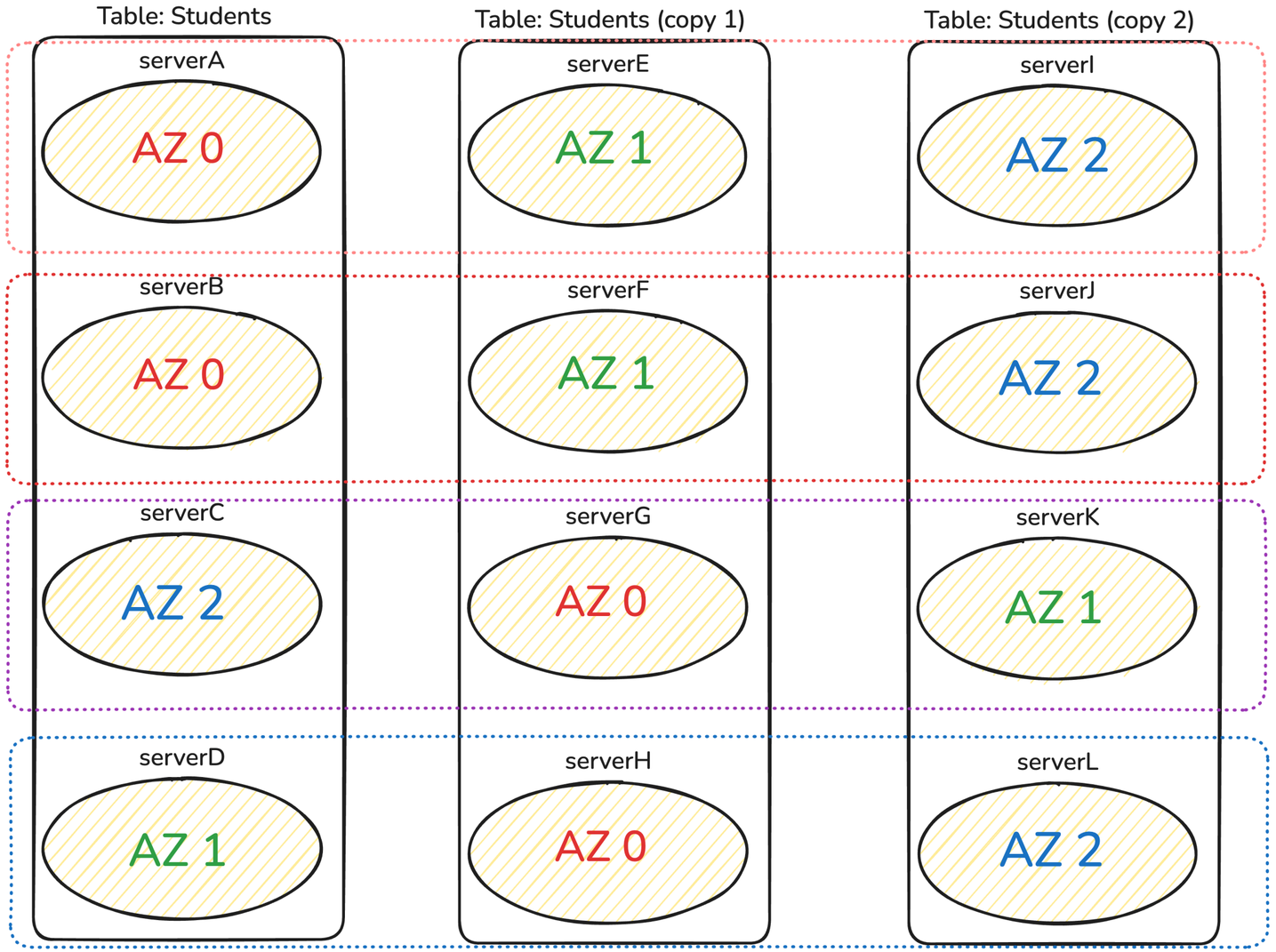
Mirror Server Set: How to place shards in hosts predictably



Adjustable Knobs for Shard Placement

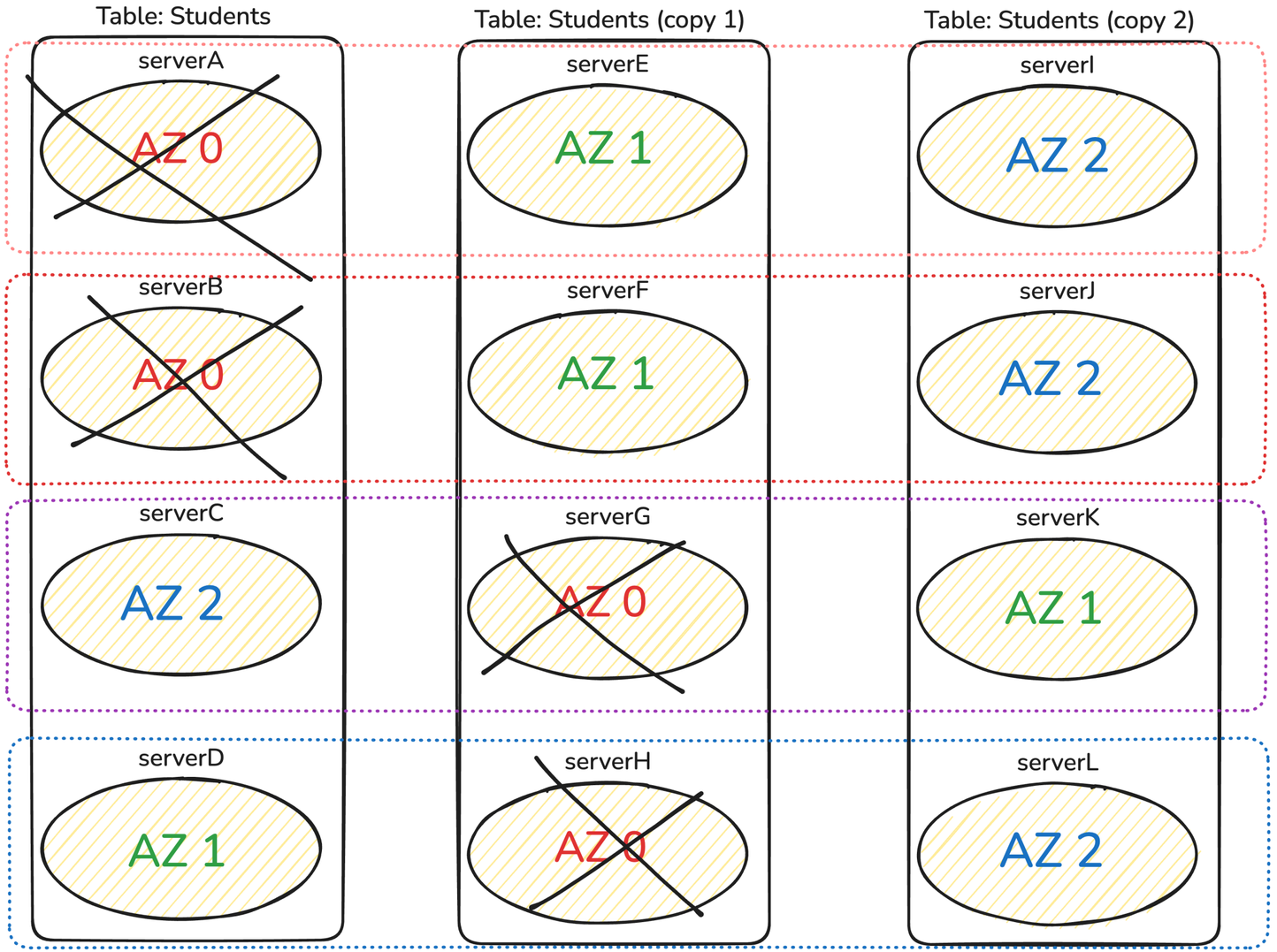


Mirror Server Set + Availability Zone Diversity



3 Availability Zones: (AZ 0 , AZ 1 , AZ 2)

Mirror Server Set + Availability Zone Diversity



Mirror server set pink:
AZ diverse

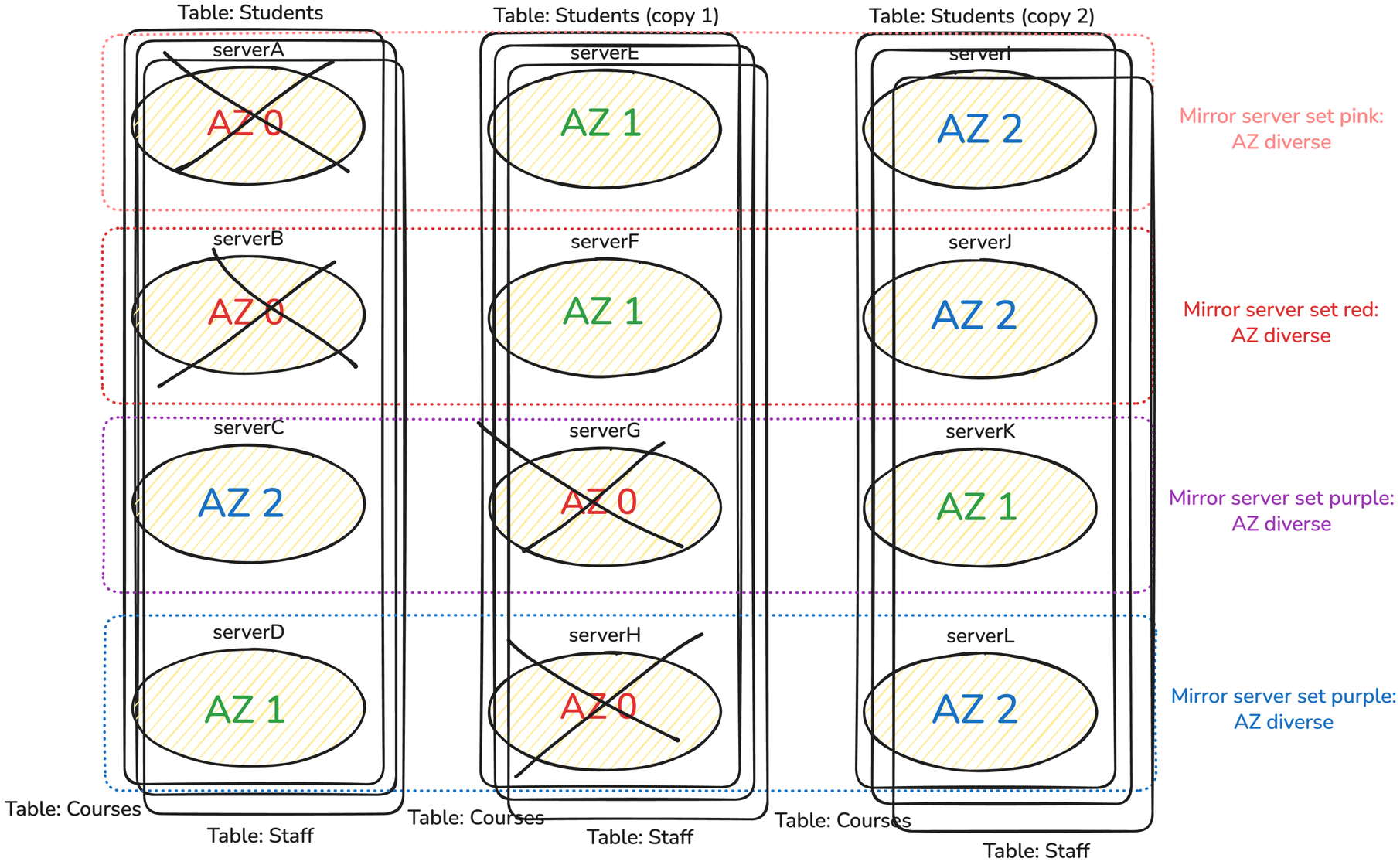
Mirror server set red:
AZ diverse

Mirror server set purple:
AZ diverse

Mirror server set purple:
AZ diverse

AZ 0 fully down!!

Mirror Server Set + Availability Zone Diversity



AZ 0 fully down!!

With AZ diversity and mirror server sets, one AZ going fully down is safe for all tables served by a cluster.

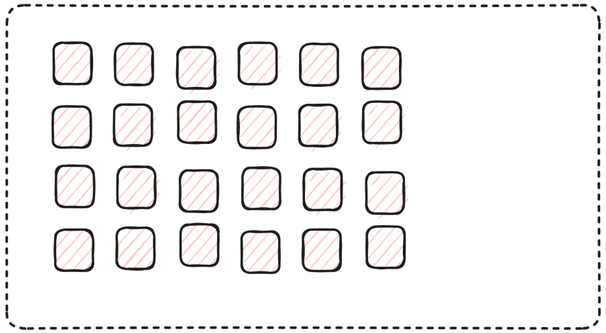
Prereq: Containerization

4

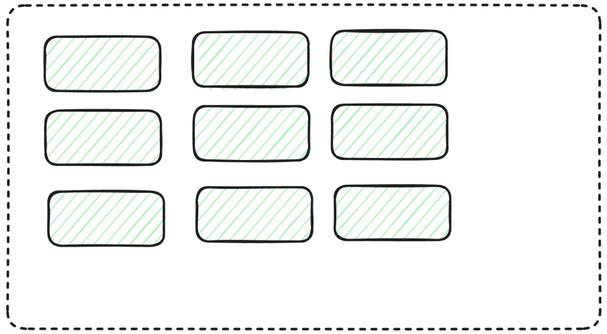
Multi-SKU Management

Manage node pools via k8s label:
pinot-server-*[RAM_SIZE]*-*[SSD_SIZE]*-SSD

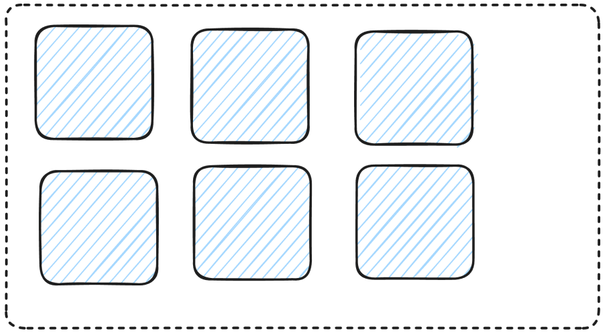
pinot-server-64GB-SSD



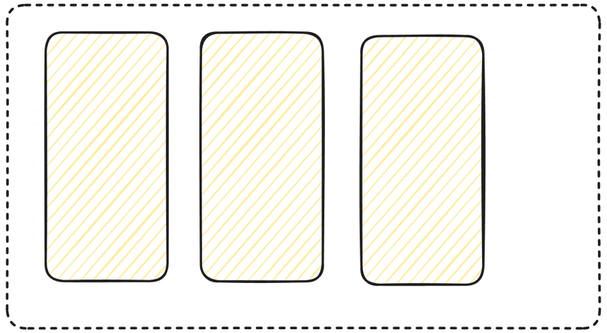
pinot-server-128GB-SSD



pinot-server-128GB-6p4t-SSD



pinot-server-128GB-12t-SSD



```
affinity:
  nodeAffinity:
    requiredDuringSchedulingIgnoredDuringExecution:
      - matchExpressions:
        - affinityMode: NodeProfileAffinity
          operator: In
          values:
            - pinot-server-64GB-SSD
```

Server Config Management

Pinot team can now focus more on Pinot core database logic rather than managing & debugging individual server configurations

Pinot specific puppet modules

- Kernel settings (e.g. vm.max_map_count)
- SSD mounting
- NFS mounting
- Nginx configuration

Alternatives

- Eliminate the needs (madvise support in code)
- Permanent Volume (PV) - local
- Permanent Volume (PV)
- Embedded in app container image

Stateful vs. Stateless Components

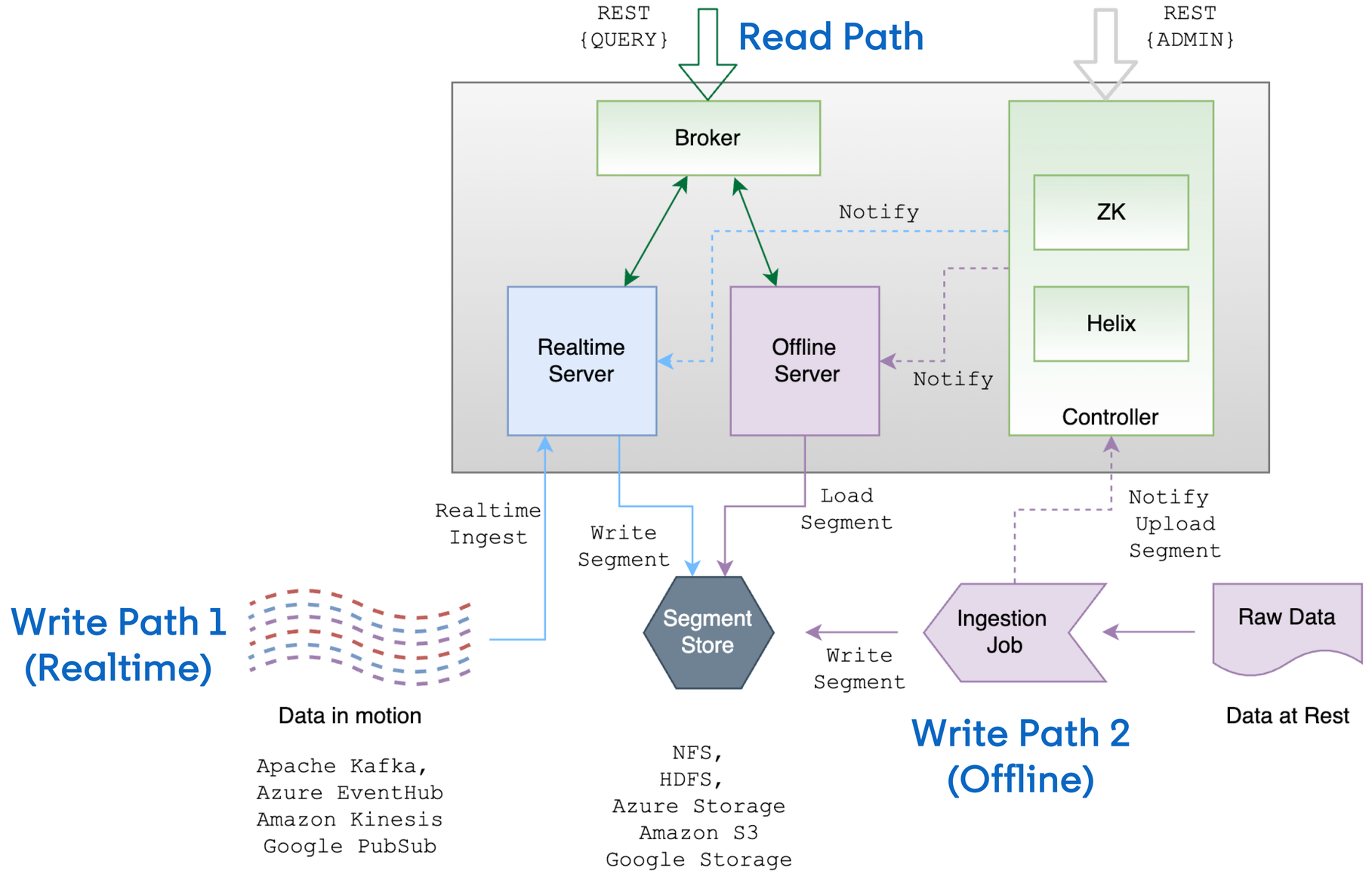
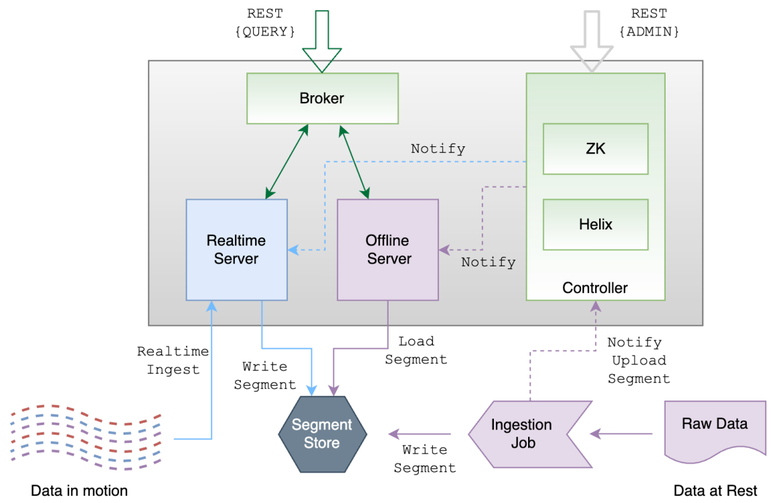


Image source: Apache Pinot doc

Stateful vs. Stateless Components



1 Broker

Accept queries from clients

Routing layer

Do NOT persist data on disk

2 Controller

Helix controller managing other Pinot components

Admin APIs

Do NOT persist data on disk

3 Realtime server

Host shards and serve queries

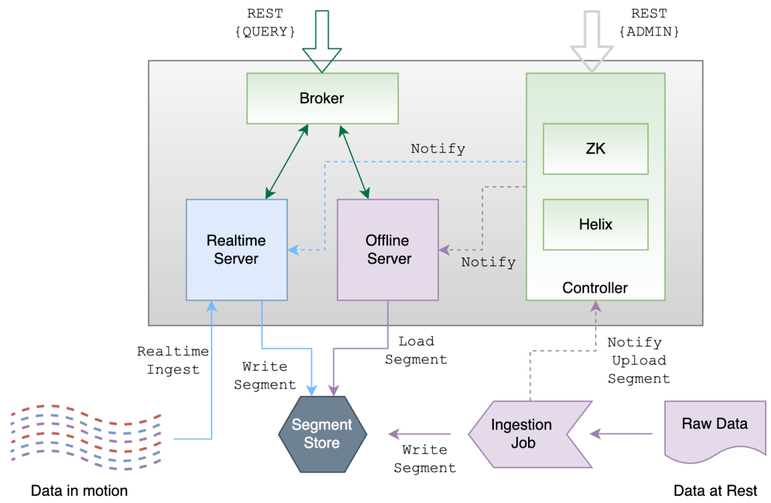
Persist data on host disk

4 Offline server

Host shards and serve queries

Persist data on host disk

Stateful vs. Stateless Components



1 Broker

Accept queries from clients

Routing layer

Do NOT persist data on disk

2 Controller

Helix controller managing other Pinot components

Admin APIs

Do NOT persist data on disk

Stateless Applications

3 Realtime server

Host shards and serve queries

Persist data on host disk

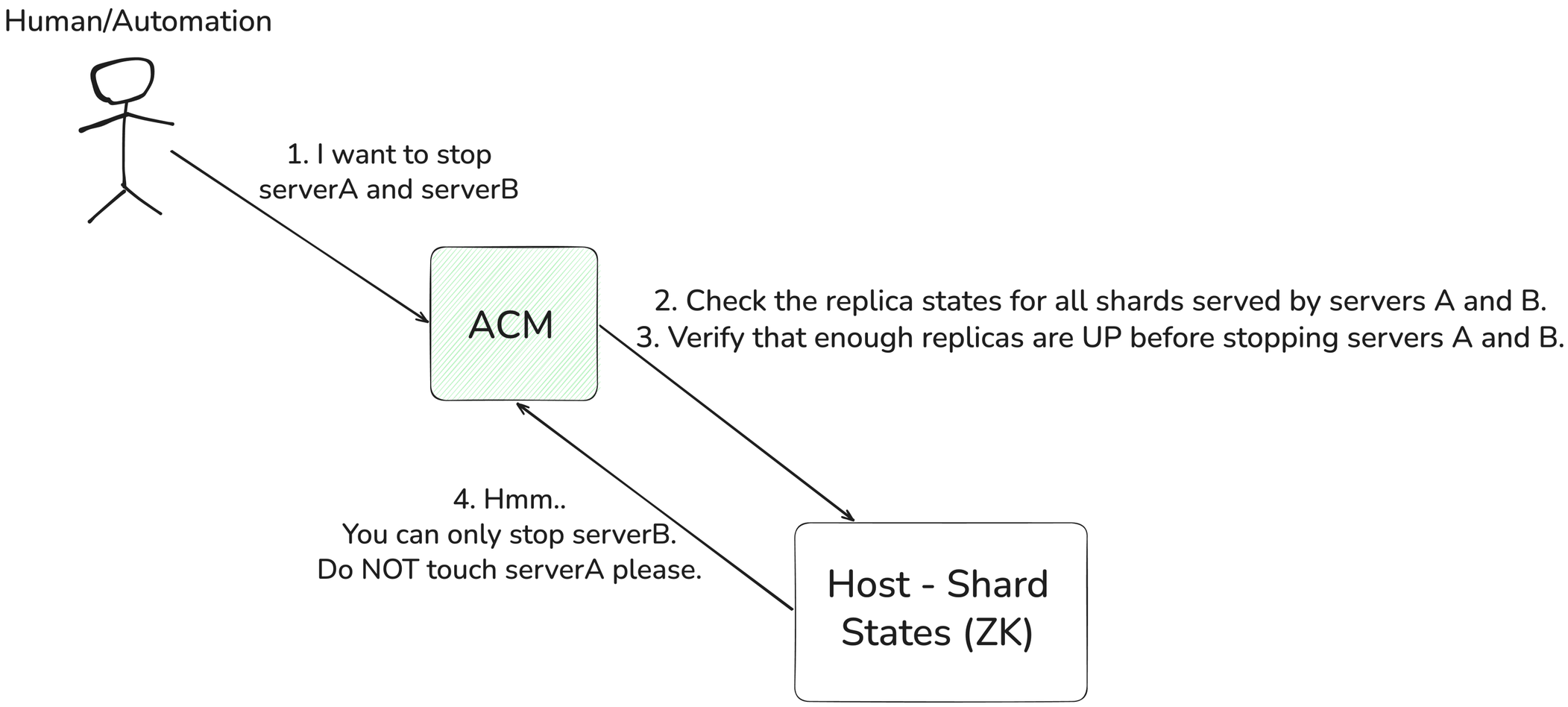
4 Offline server

Host shards and serve queries

Persist data on host disk

Stateful Applications

Host Maintenance – Application Cluster Manager (ACM)

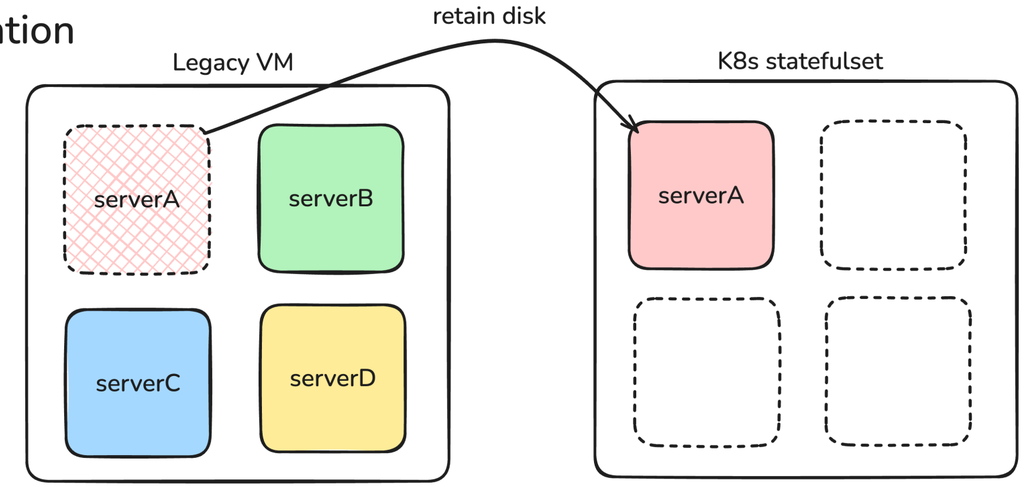


Migration Orchestrator: Design

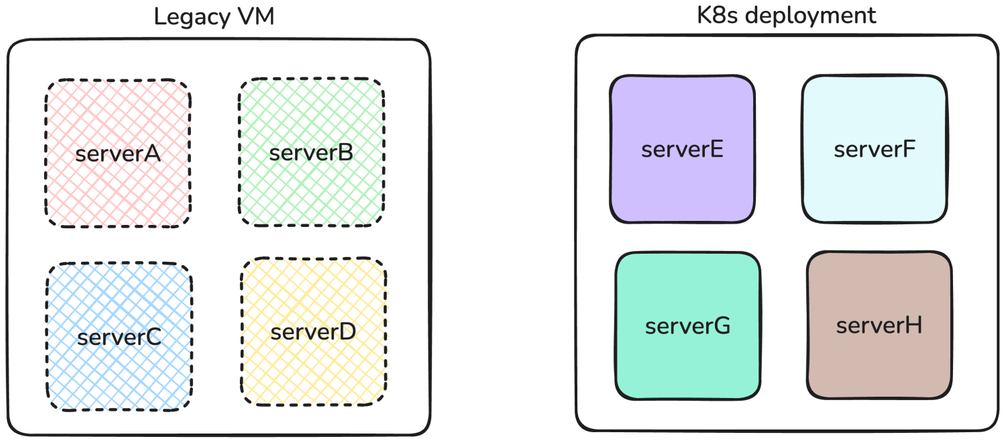
5

In-Place vs. Out-of-Place Migrations

In-Place Migration



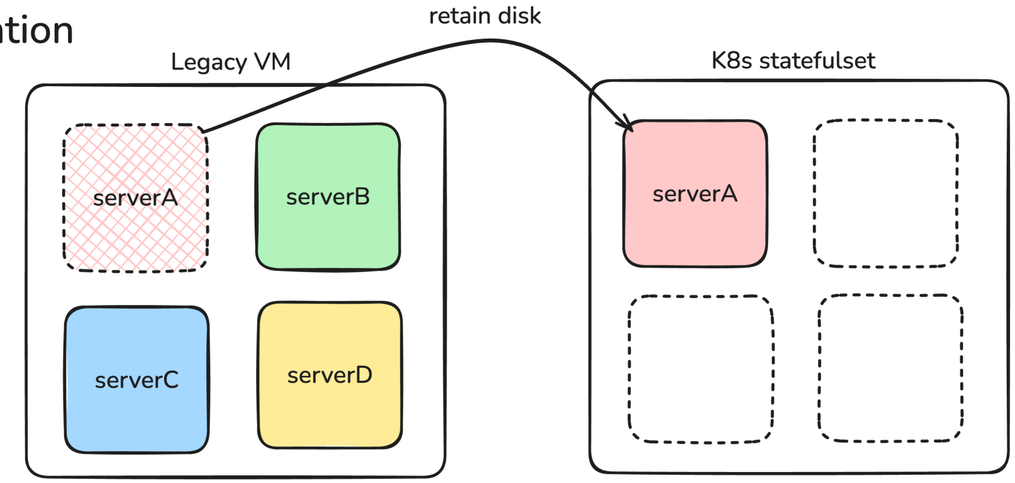
Out-of-Place Migration



take down serverA, B, C, and D after the new set of servers are UP in the target K8s deployment

In-Place vs. Out-of-Place Migrations

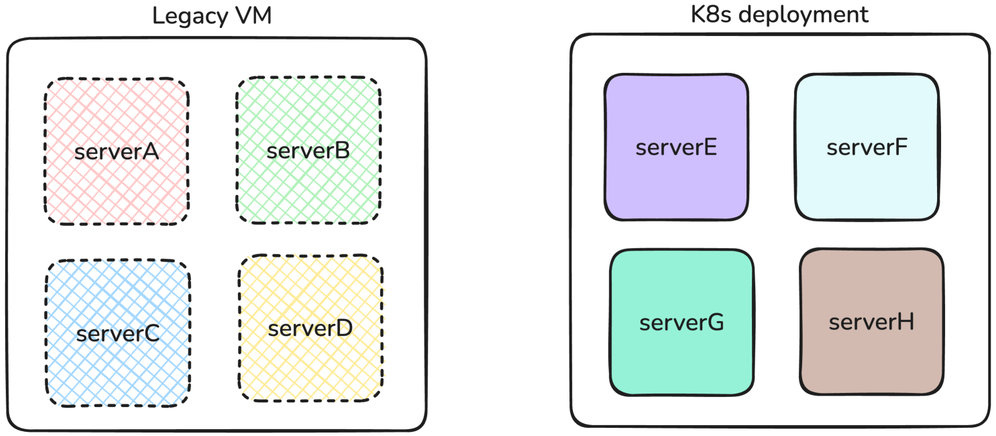
In-Place Migration



+ Don’t require spare HW
+ Don’t require data movement
- Slow
- Tricky to handle bad hosts during migration

Stateful Applications

Out-of-Place Migration

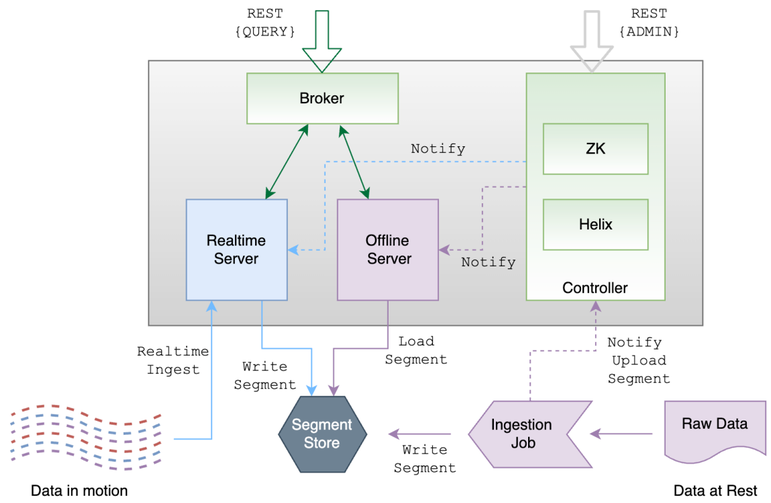


take down serverA, B, C, and D after the new set of servers are UP in the target K8s deployment

+ Fast
+ Handling bad hosts is easy (replace)
- Require spare HW
- Require data movement (if data is stored on hosts)

Stateless Applications

Stateful vs. Stateless Components



1 **Broker**

2 **Controller**

3 **Realtime server**

4 **Offline server**

Stateless Applications
-> Out-of-Place Migration

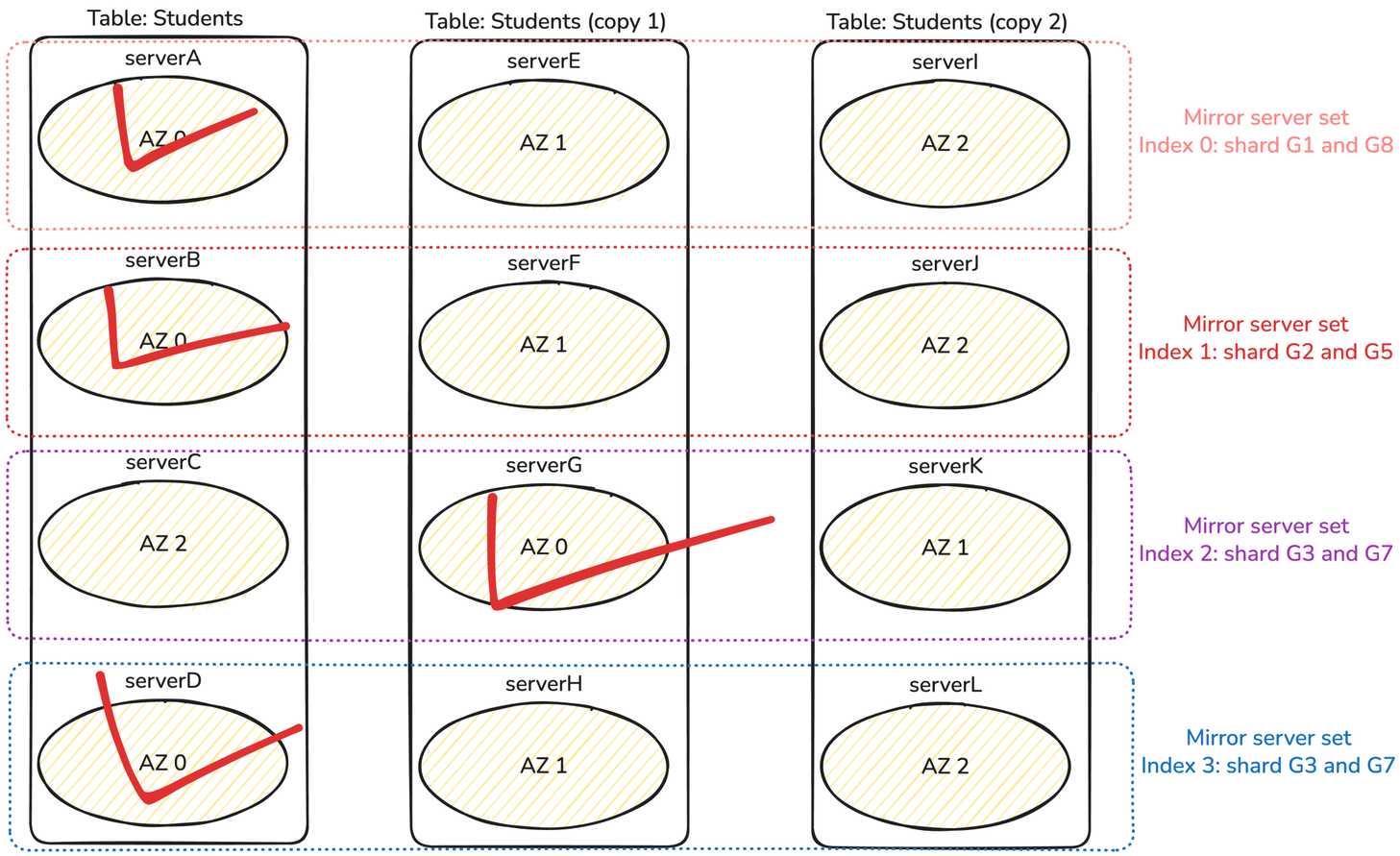
Stateful Applications
-> In-Place Migration

How do you ensure safety and performance during migration?

A very reasonable person asked

In-Place Migration: Safety

Availability Zone-Aware Shard Placement from part 3 to the rescue!



Migrate all hosts in AZ 0: serverA, serverB, serverG, serverD

Migrate all hosts in AZ 1: serverE, serverF, serverK, serverH

...

Automated Performance Validation

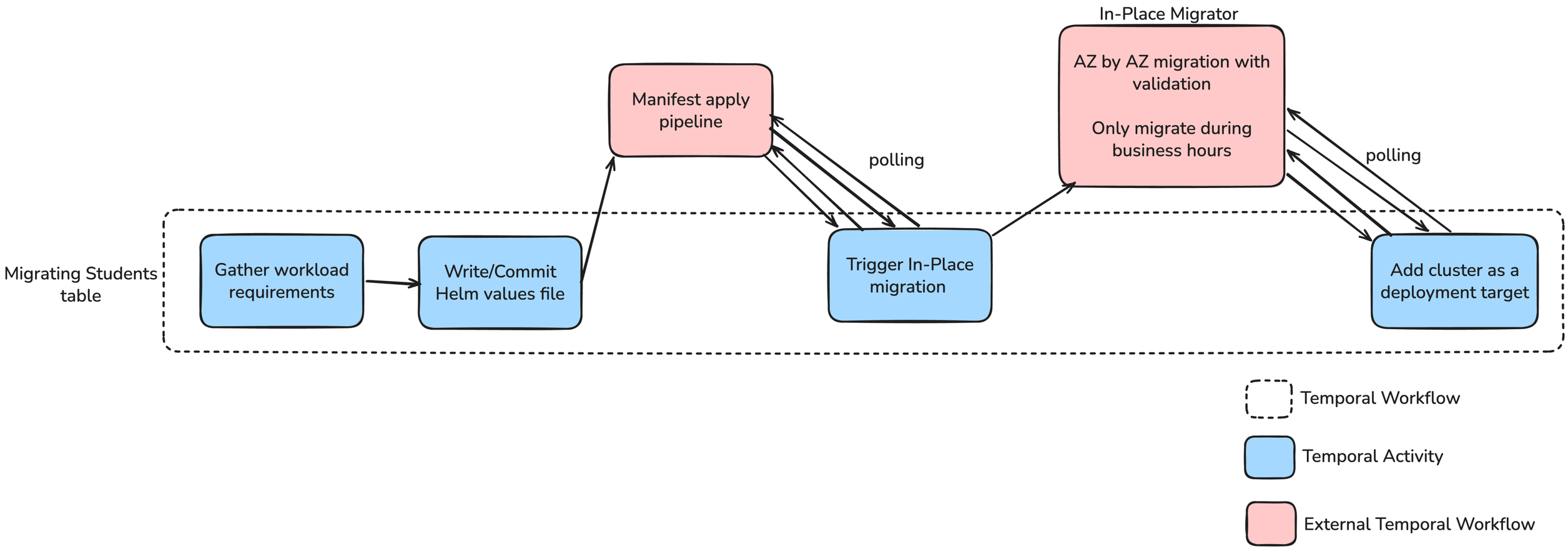
Laundry list of automated validation checks for each host migrated

- Is the host healthy?
- Log validation
  - Are there any new errors?
  - Is the host ingesting data?
- Metric validation (compared to existing hosts)
  - Is the query latency in expected range?
  - Is CPU usage in expected range?

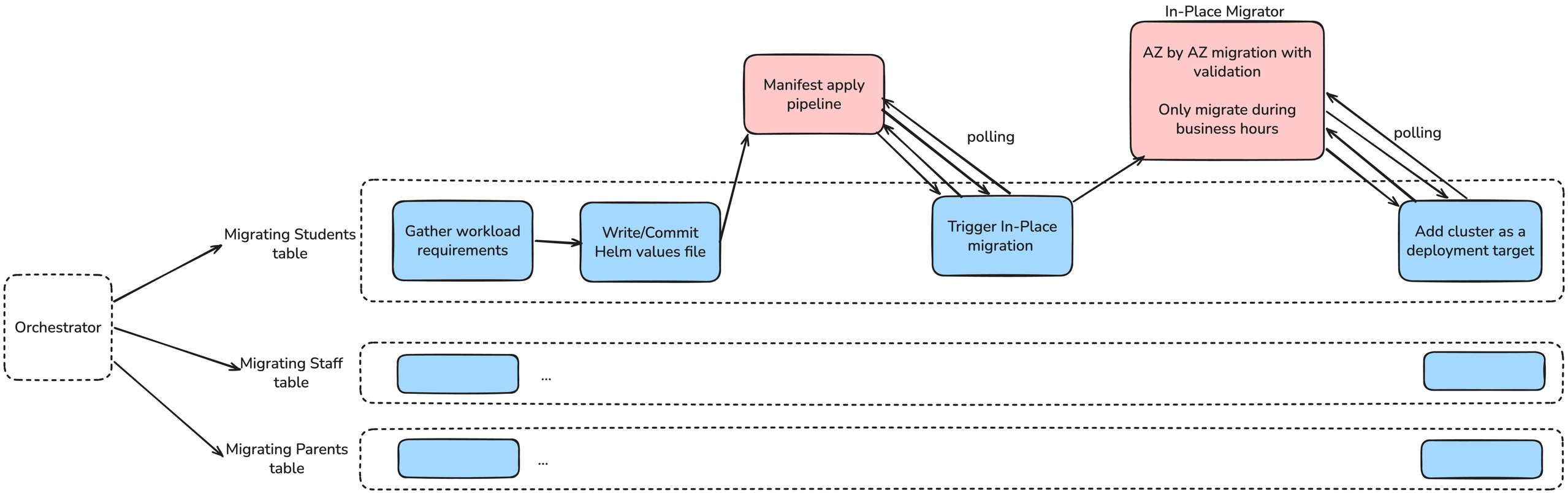
Migration Orchestrator: Execution

6

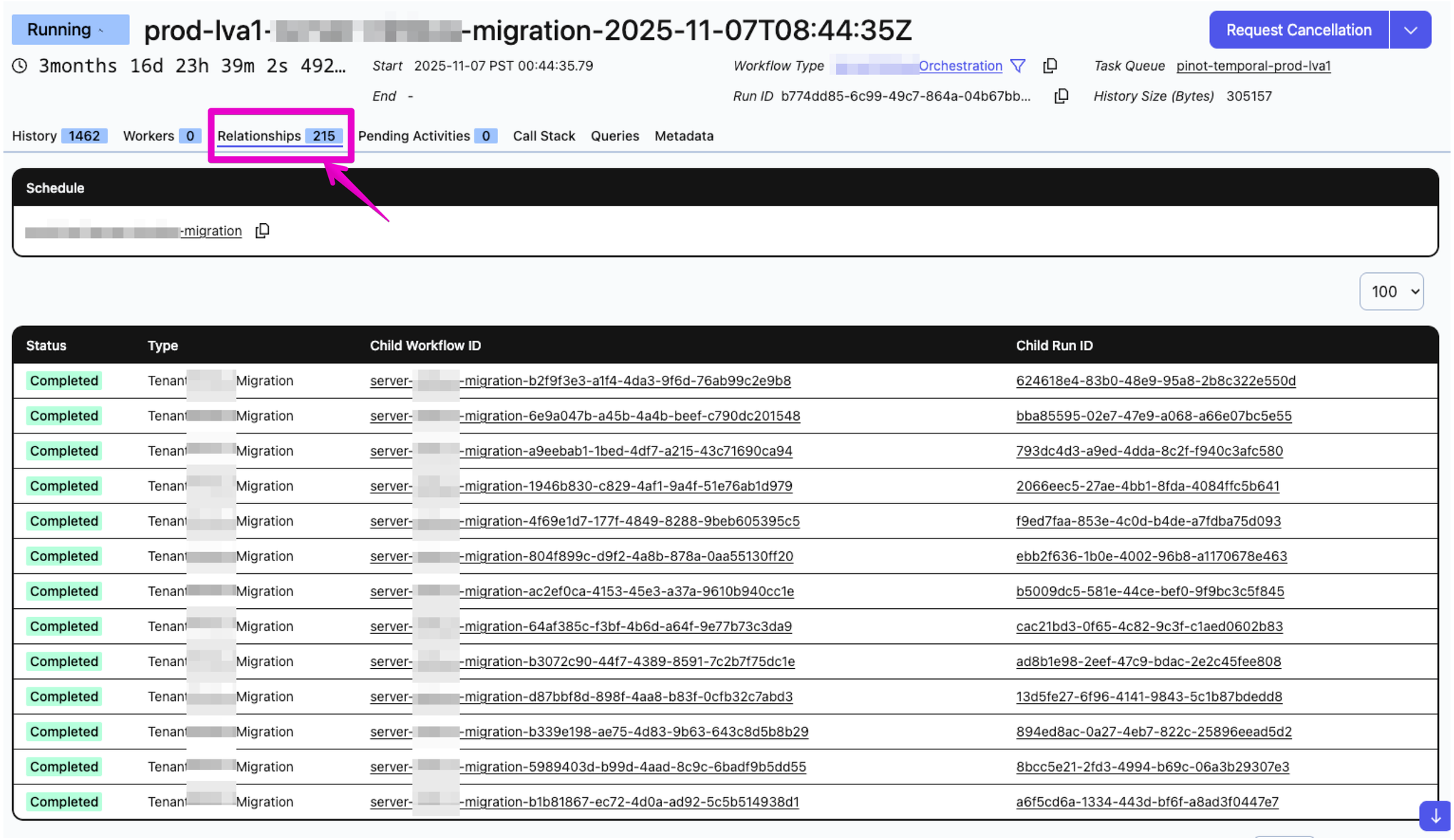
Temporal Workflow for In-Place Migration



Temporal Workflow for In-Place Migration



Temporal Workflow Used



Lessons Learned & Futures

Lessons

You will save migration time by spending more time preparing for the migration.

Be careful when deploying during the migration.

Futures

Pinot pod stacking support

K8s operator to manage Pinot clusters

Thank you.

Special thanks to:

- Dino, Ke, Jia
- Shraddha, Siddharth, Amar

linkedin.com/in/tony-song-dev/