

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

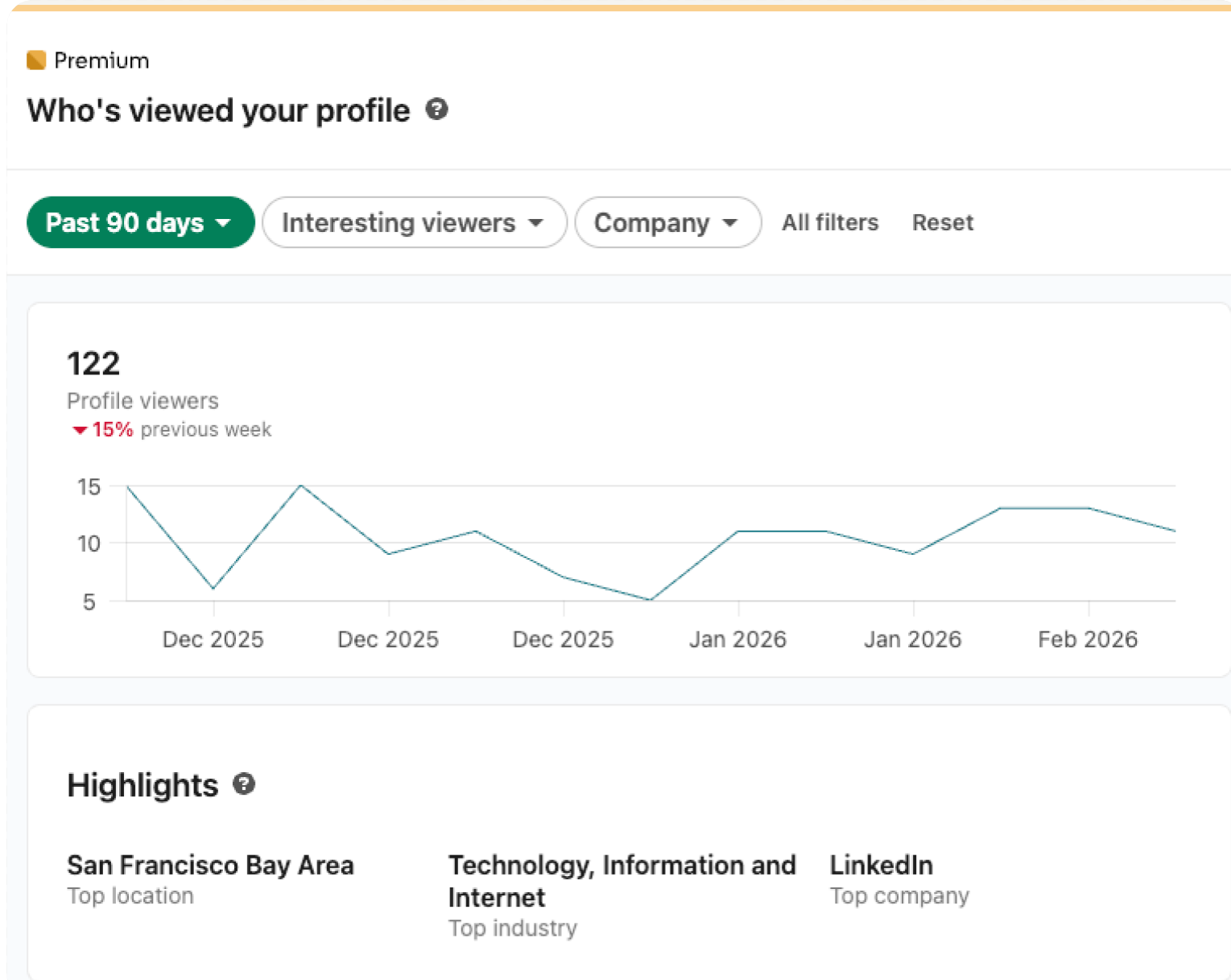


**Real-time distributed
OLAP datastore built for
low-latency, high-
throughput analytics.**

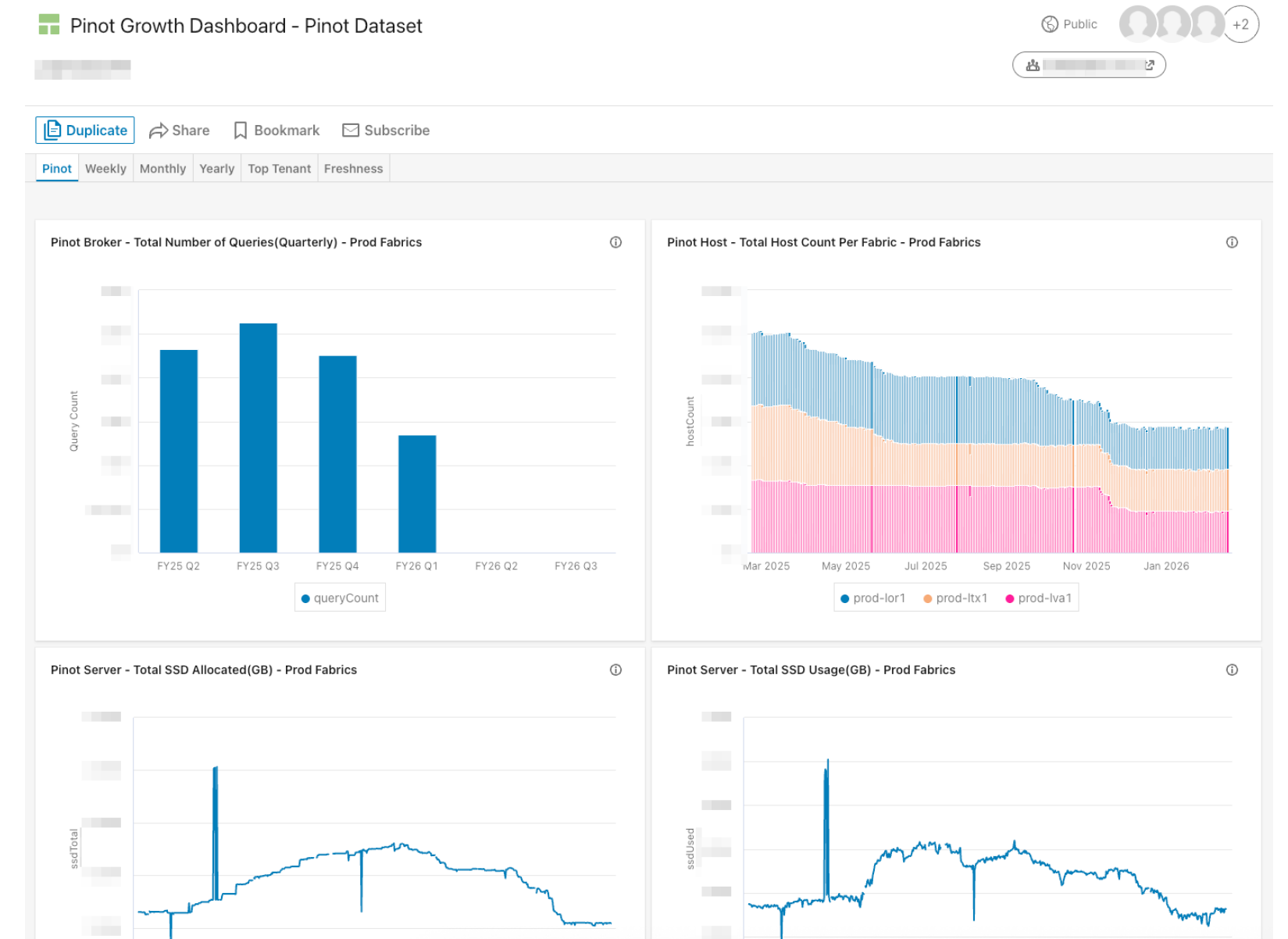


Apache Pinot @LinkedIn

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



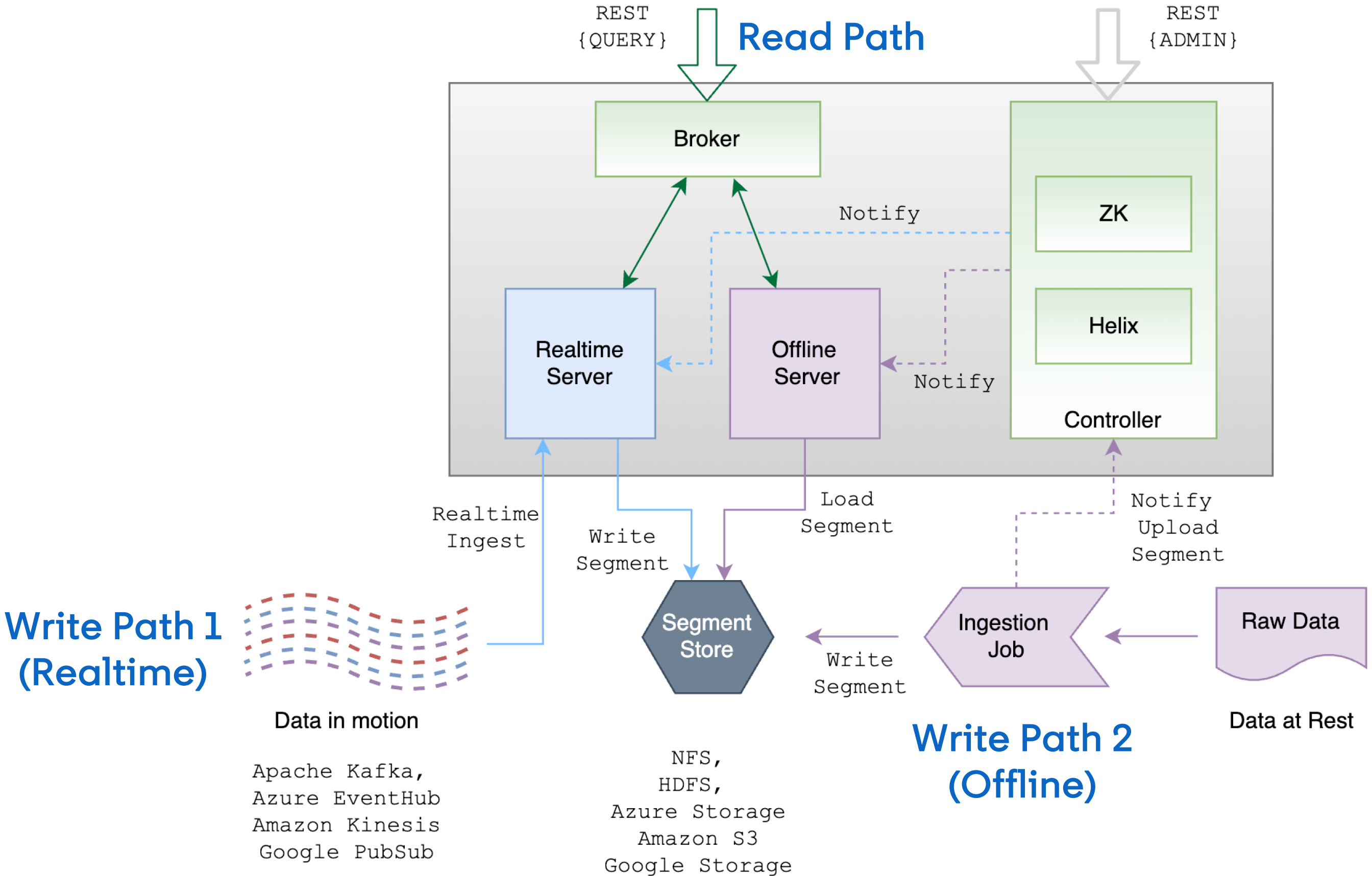
Member Facing & Enterprise Analytics



Internal Products & BI



Pinot Architecture



Apache Pinot @LinkedIn - Scale

300K+

**Queries Per
Second**

10,000+

**Serving
Hosts**

2,000+

**Broker
Hosts**

4,500+

**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?



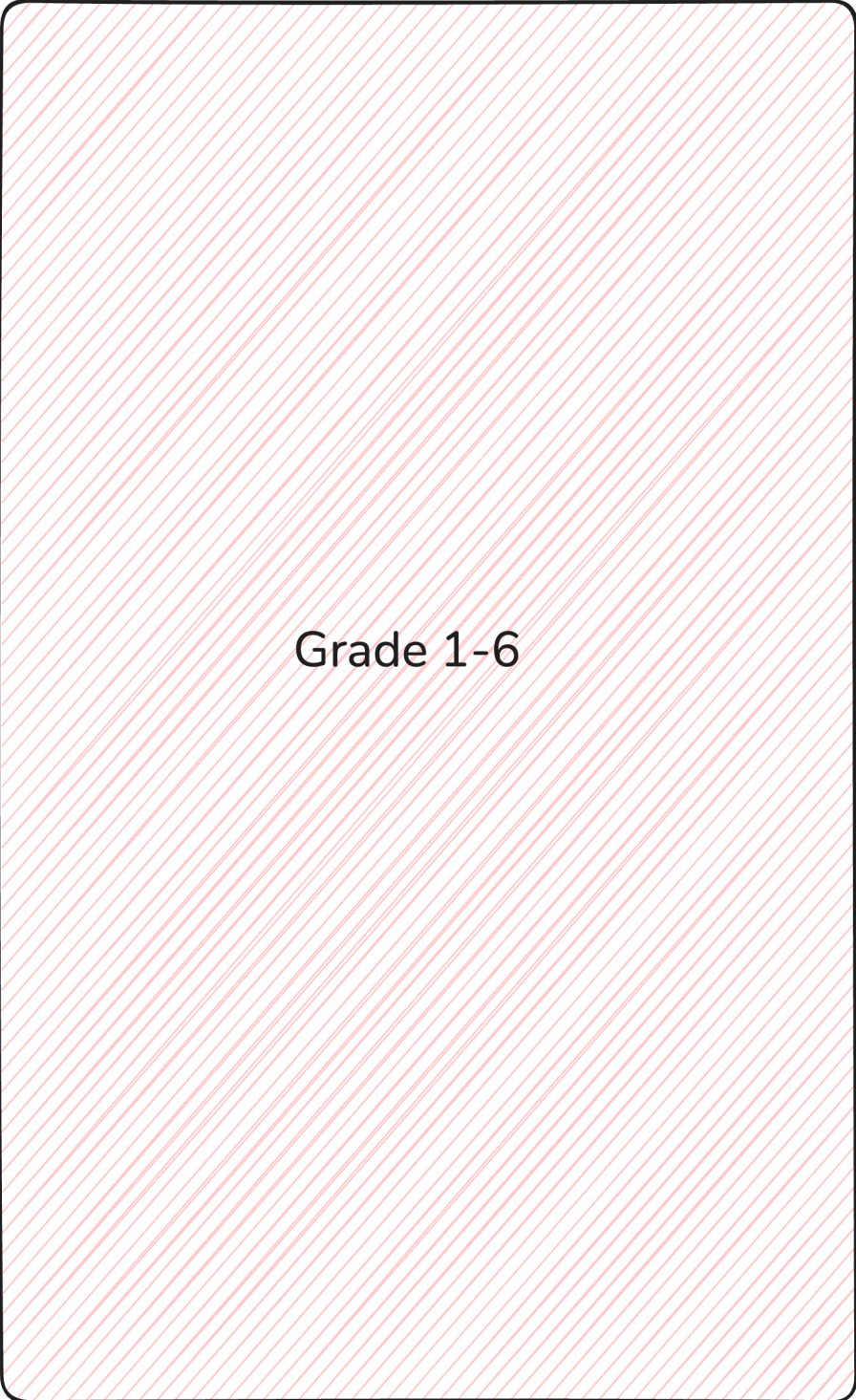
Background



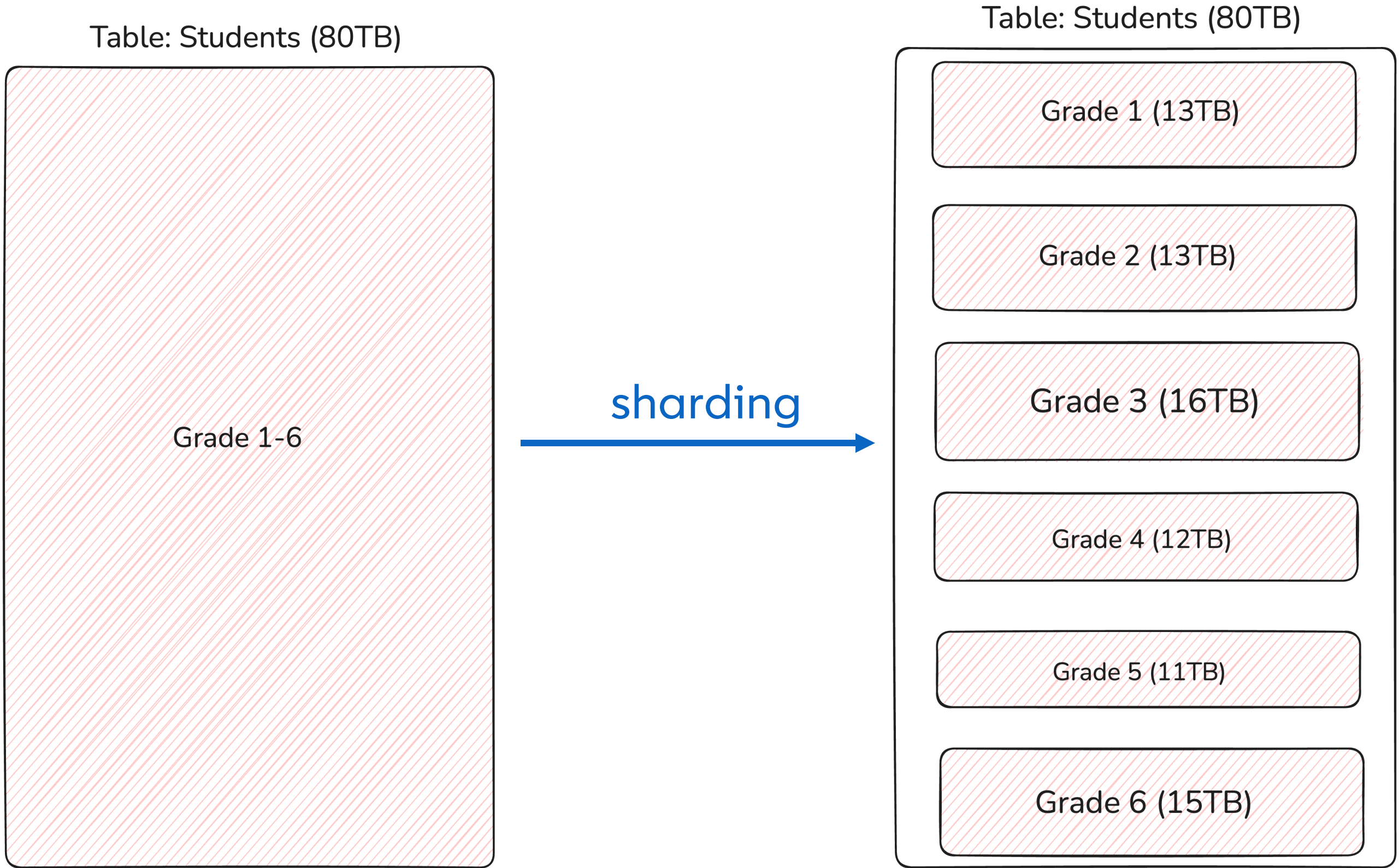
2

Shards and Replicas

Table: Students (80TB)

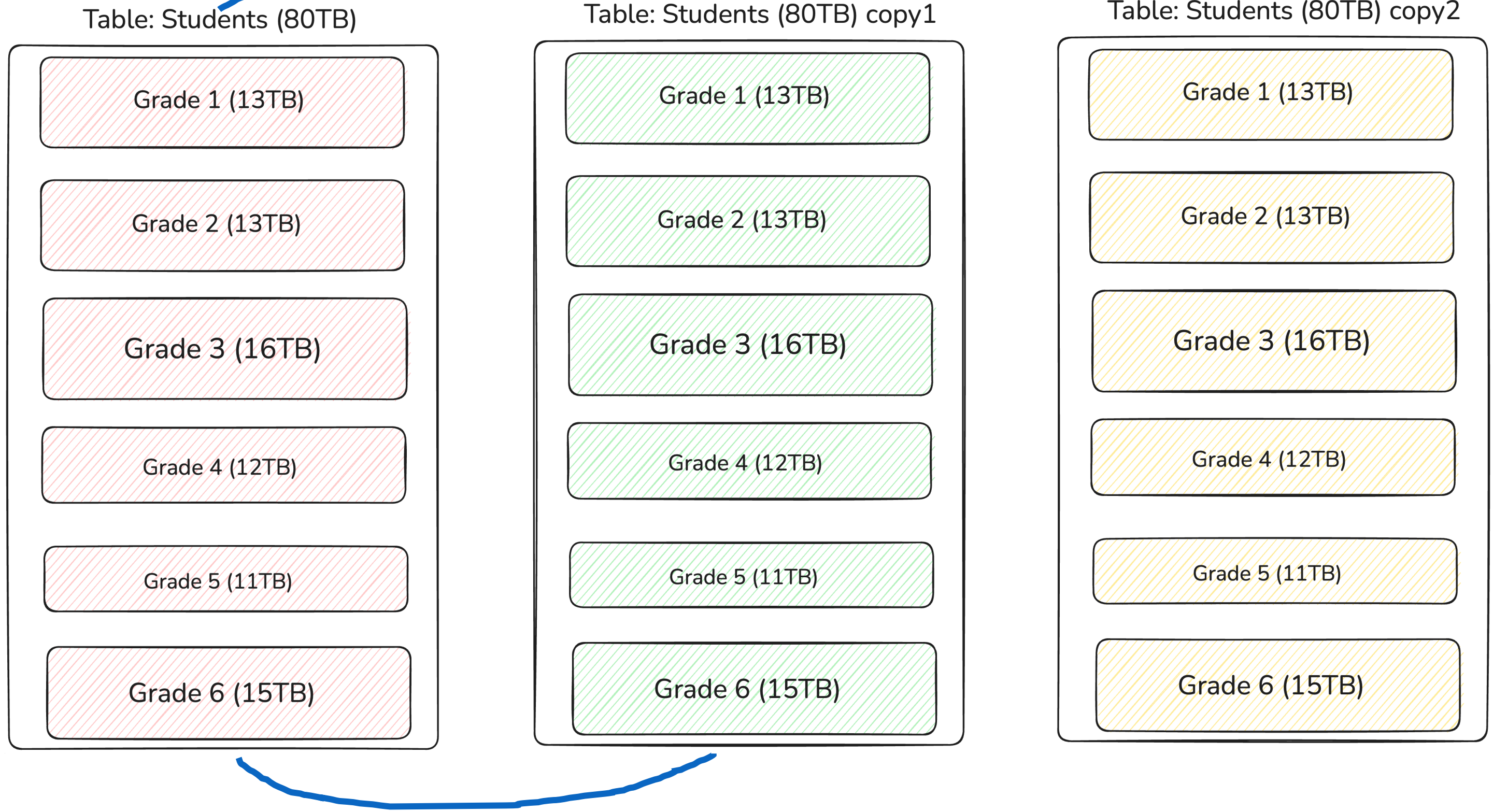


Shards and Replicas



Shards and Replicas

Replicating



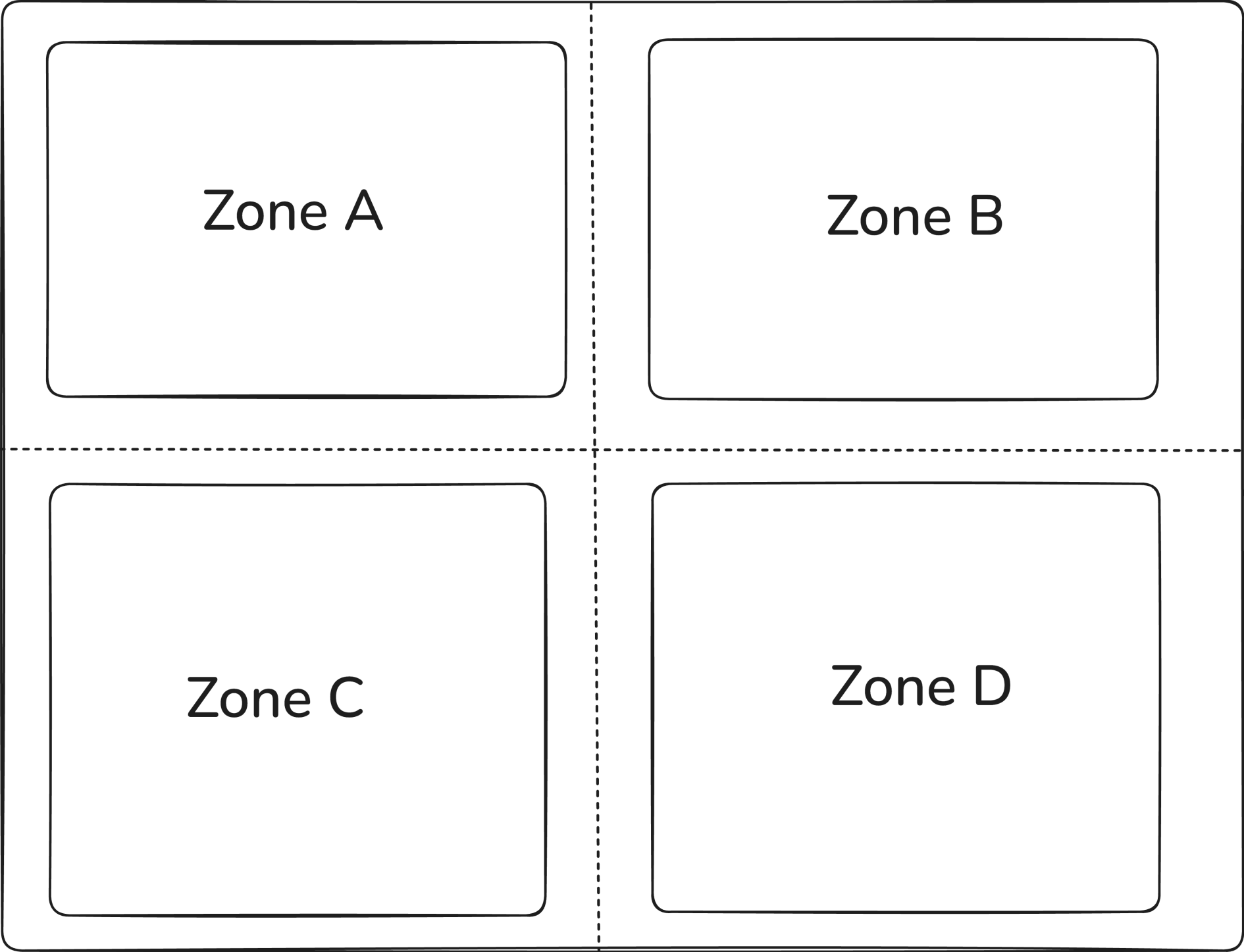
Prereq: Availability Zone- Aware Shard Placement



3

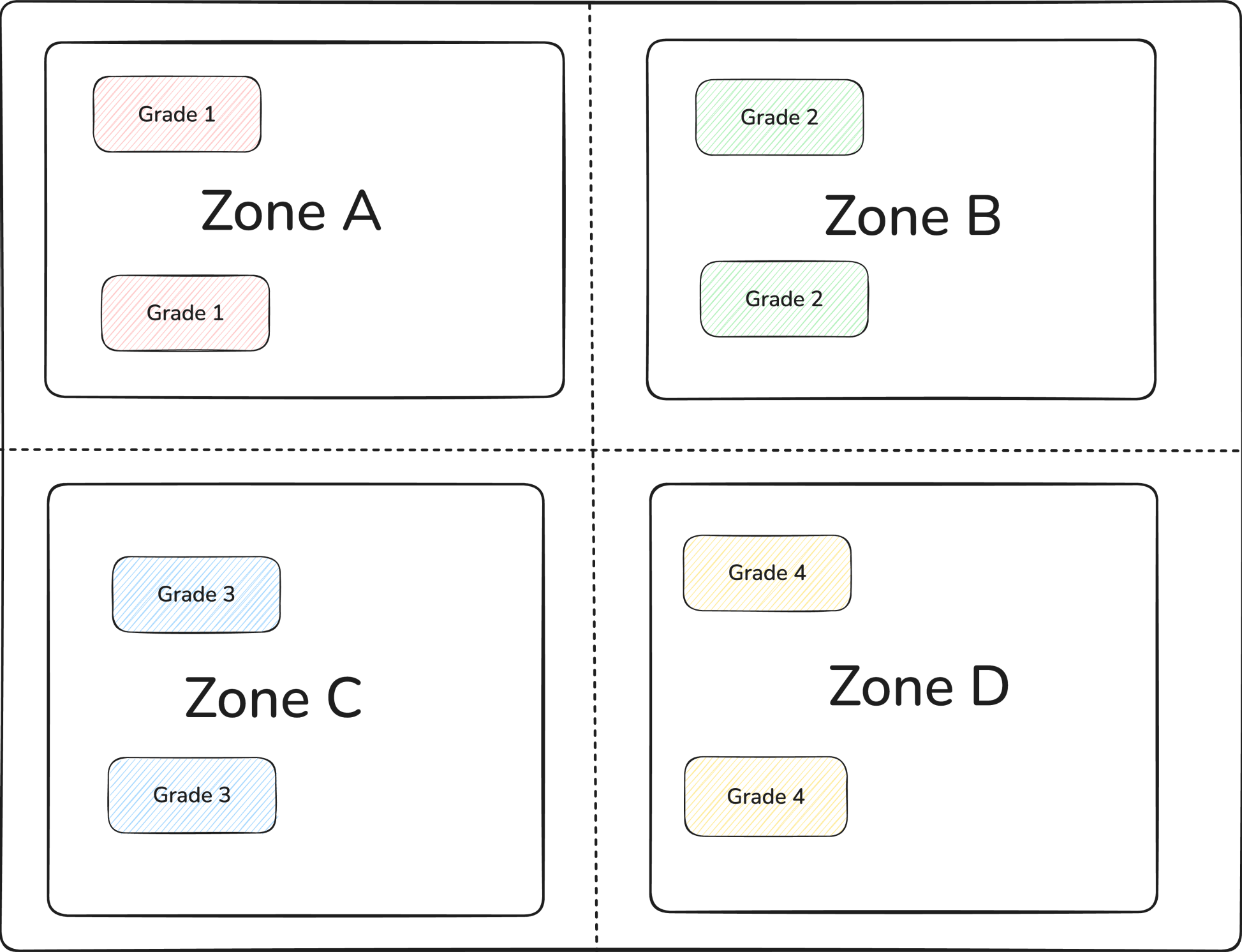
Availability Zones

Data Center in CA



Shard Placement in Availability Zones

Data Center in CA

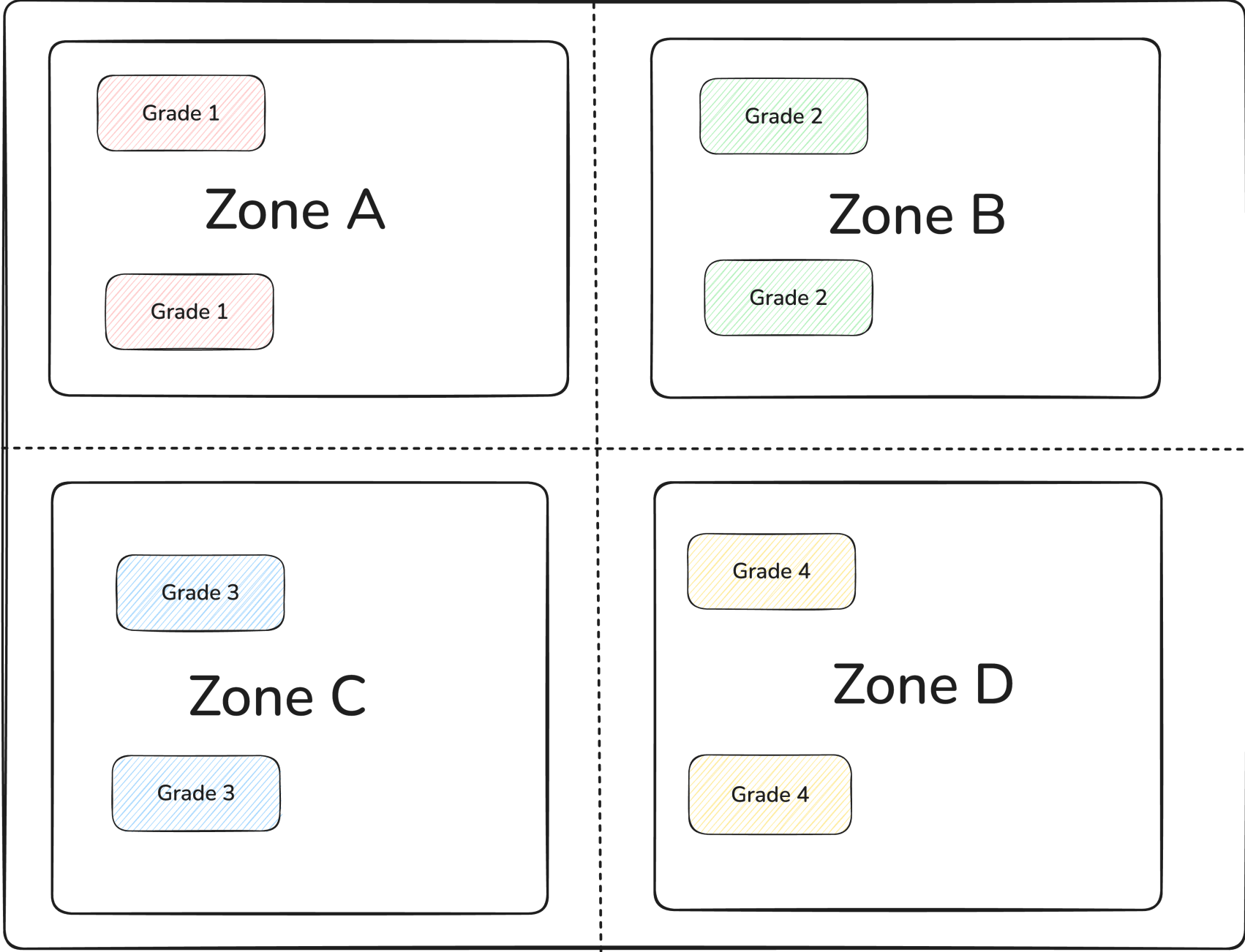


4 grades
With 2 replicas each



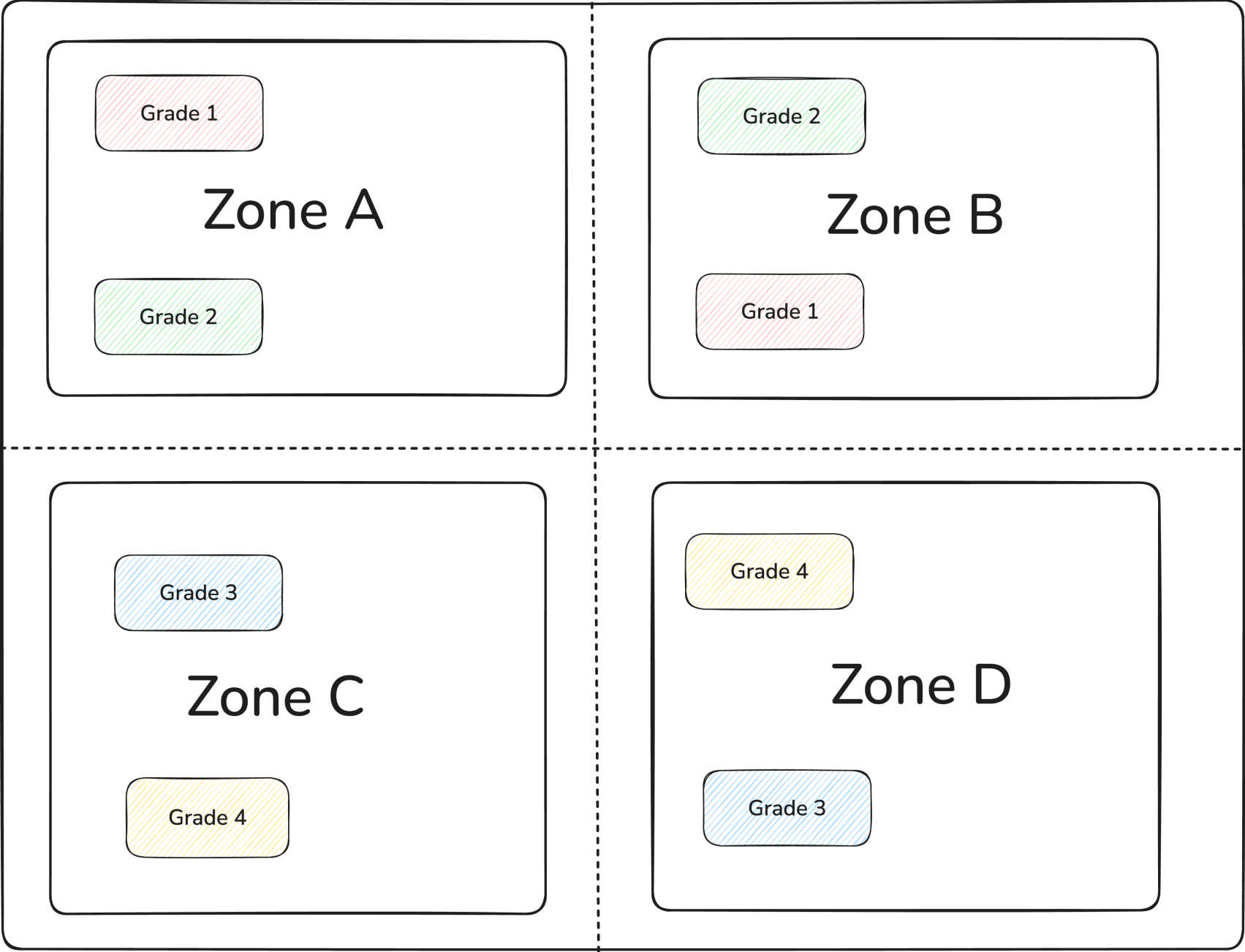
Shard Placement in Availability Zones

Data Center in CA



Zone non-diverse shard placement

Data Center in CA

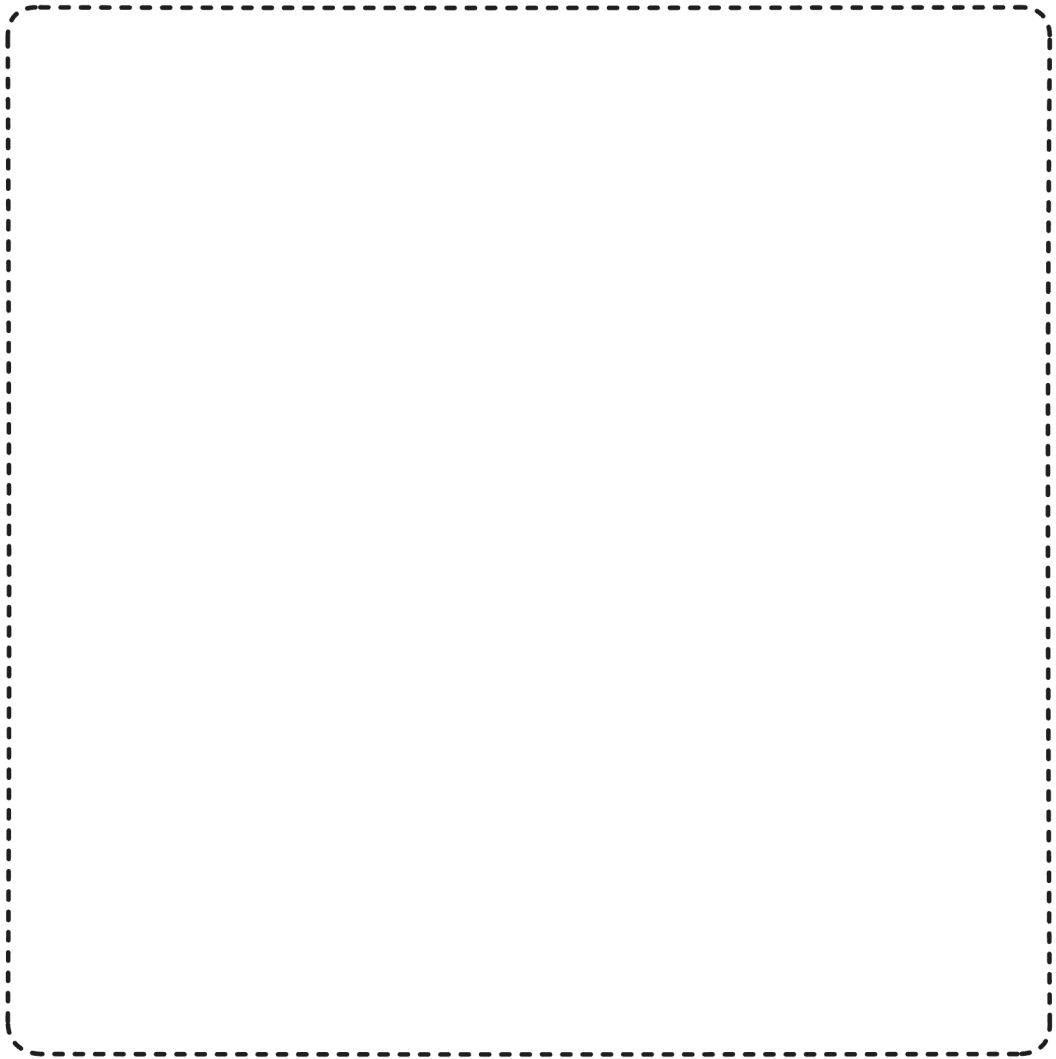


Zone diverse shard placement

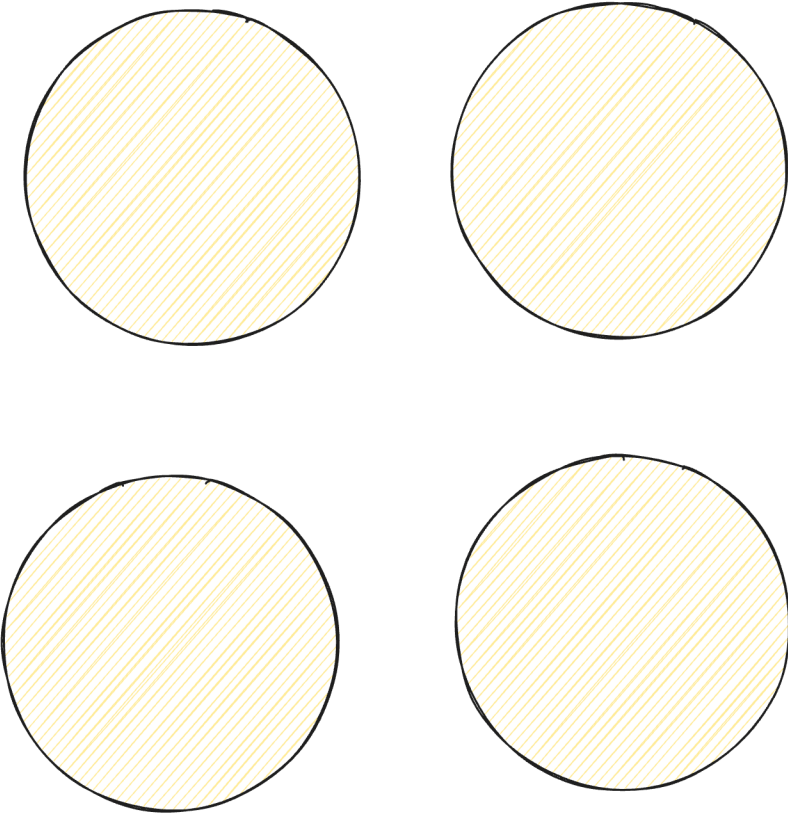
Shards: 10K+
Replicas: 40+
Availability Zones: 10+

...for a big Pinot table

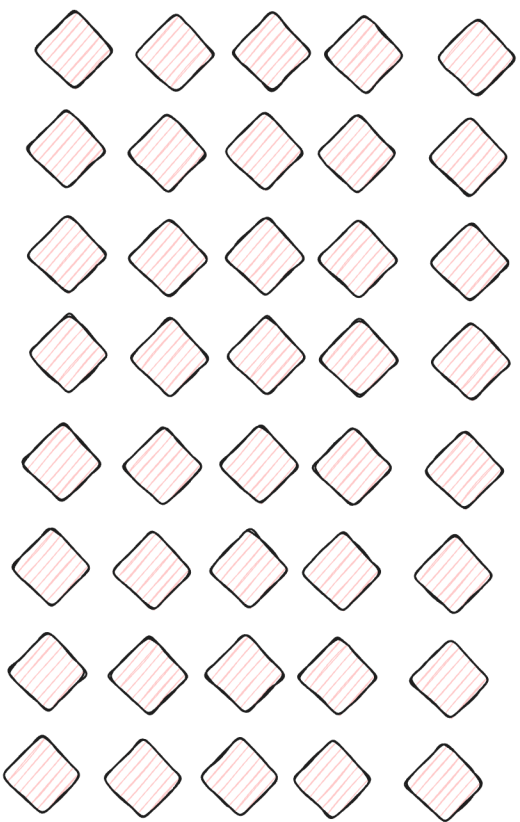
Adjustable Knobs for Shard Placement



Availability Zone

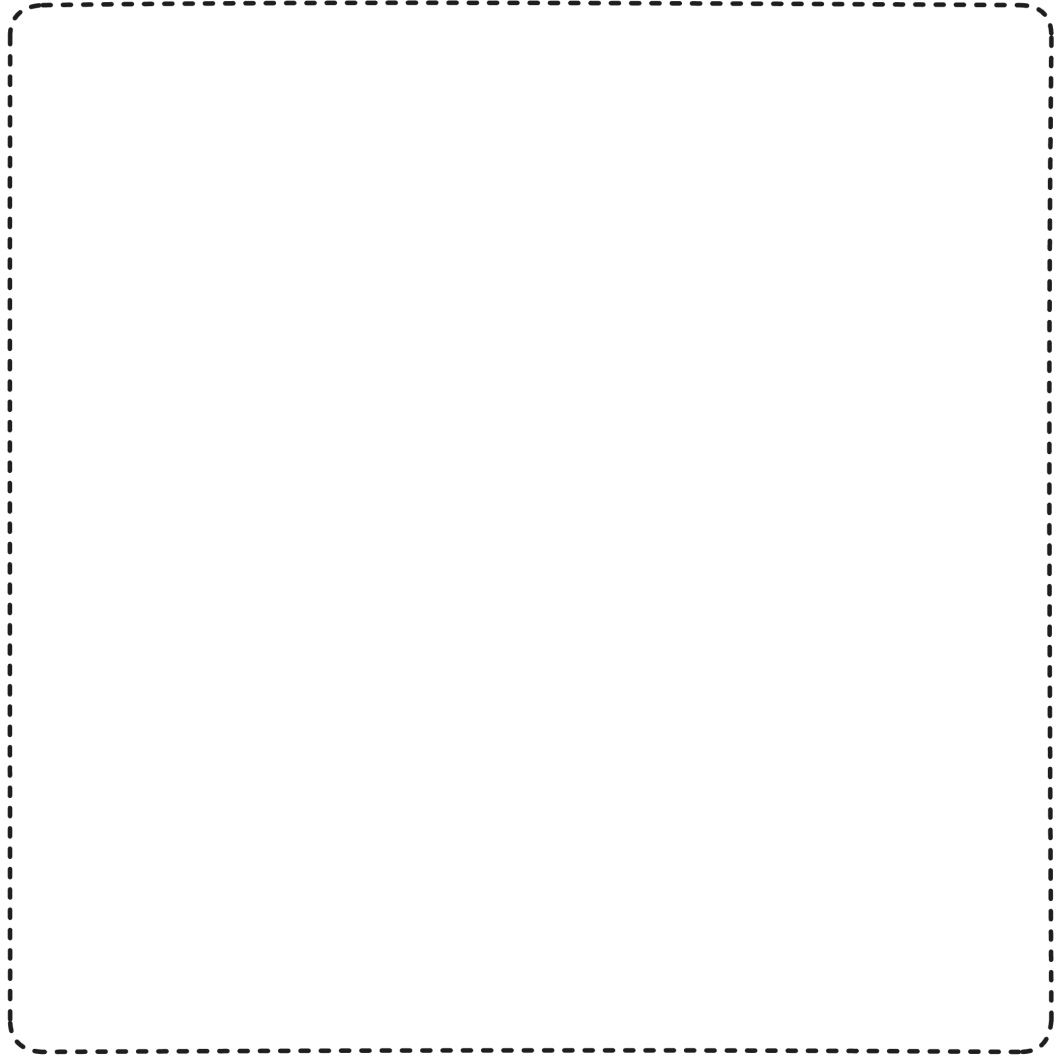


Hosts

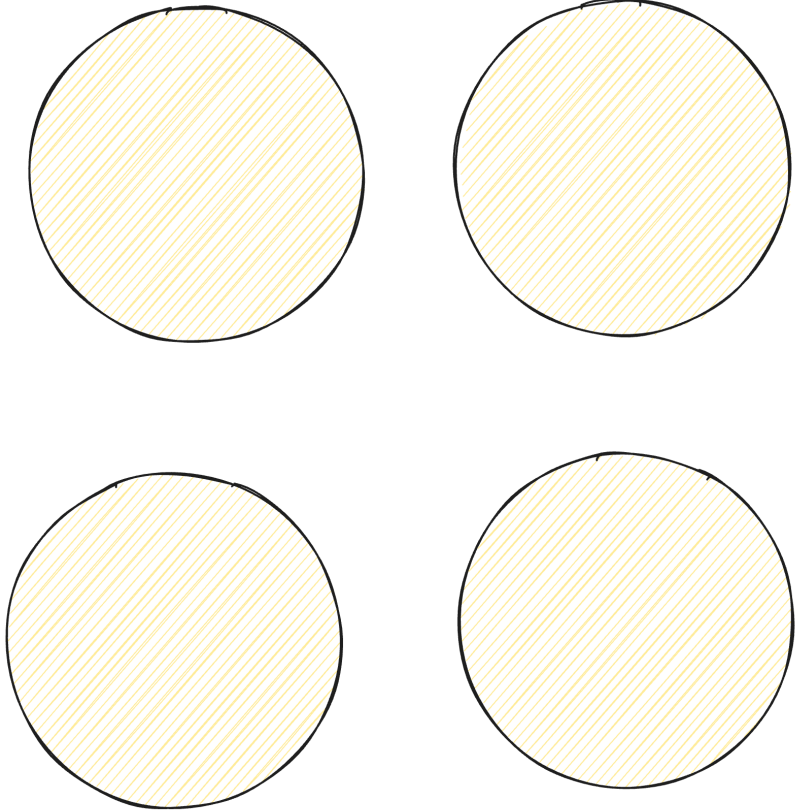


Shards

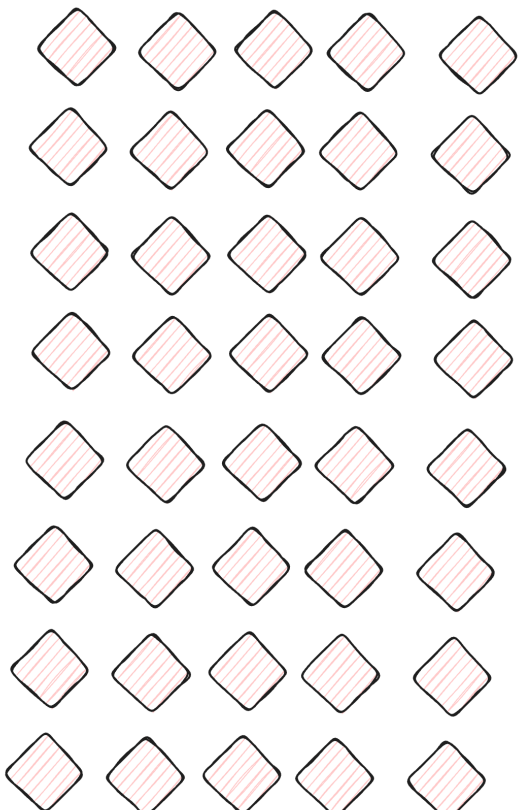
Adjustable Knobs for Shard Placement



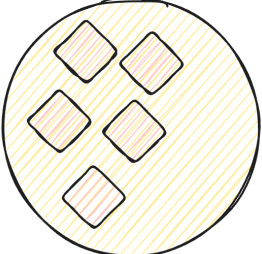
Availability Zone



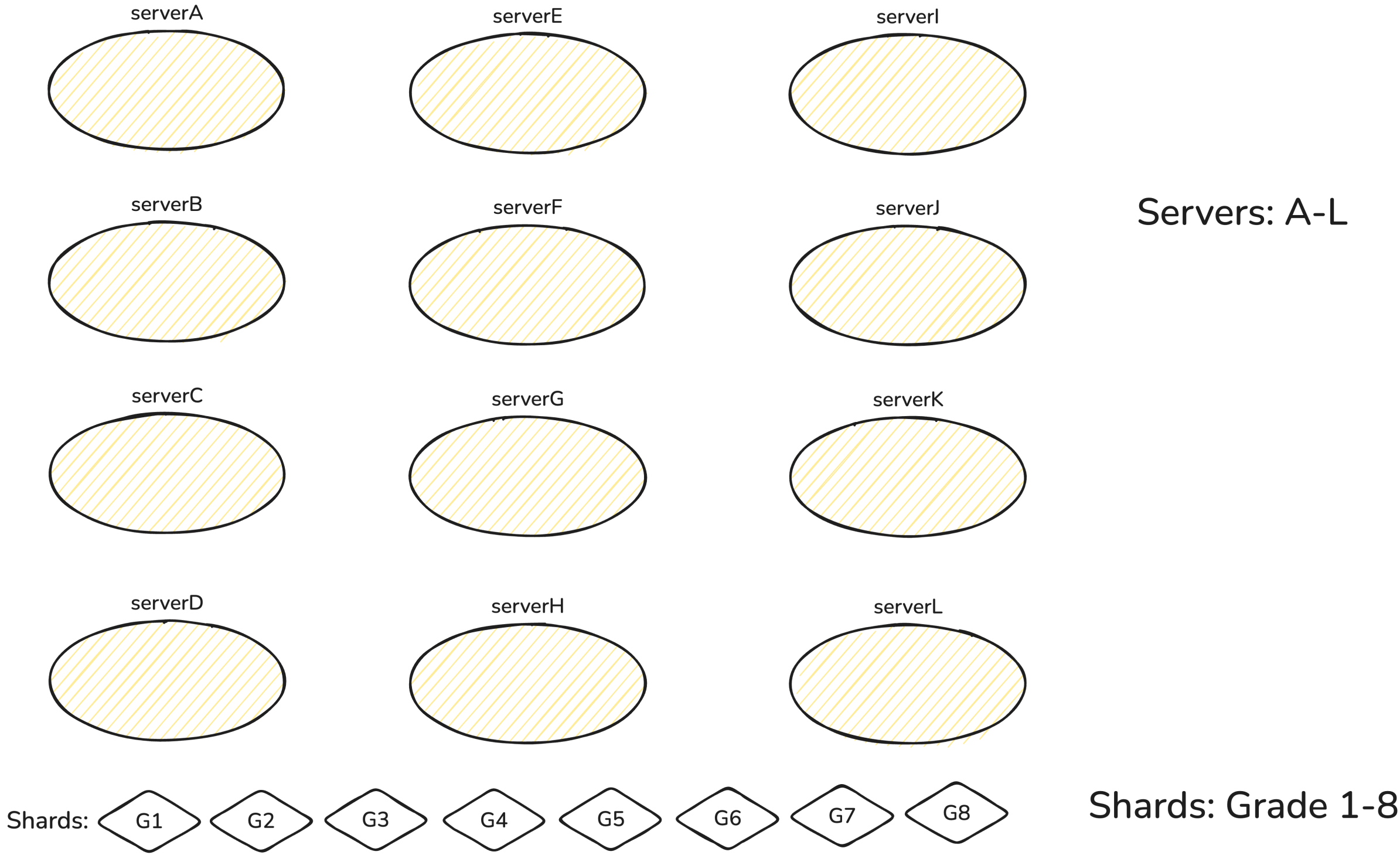
Hosts



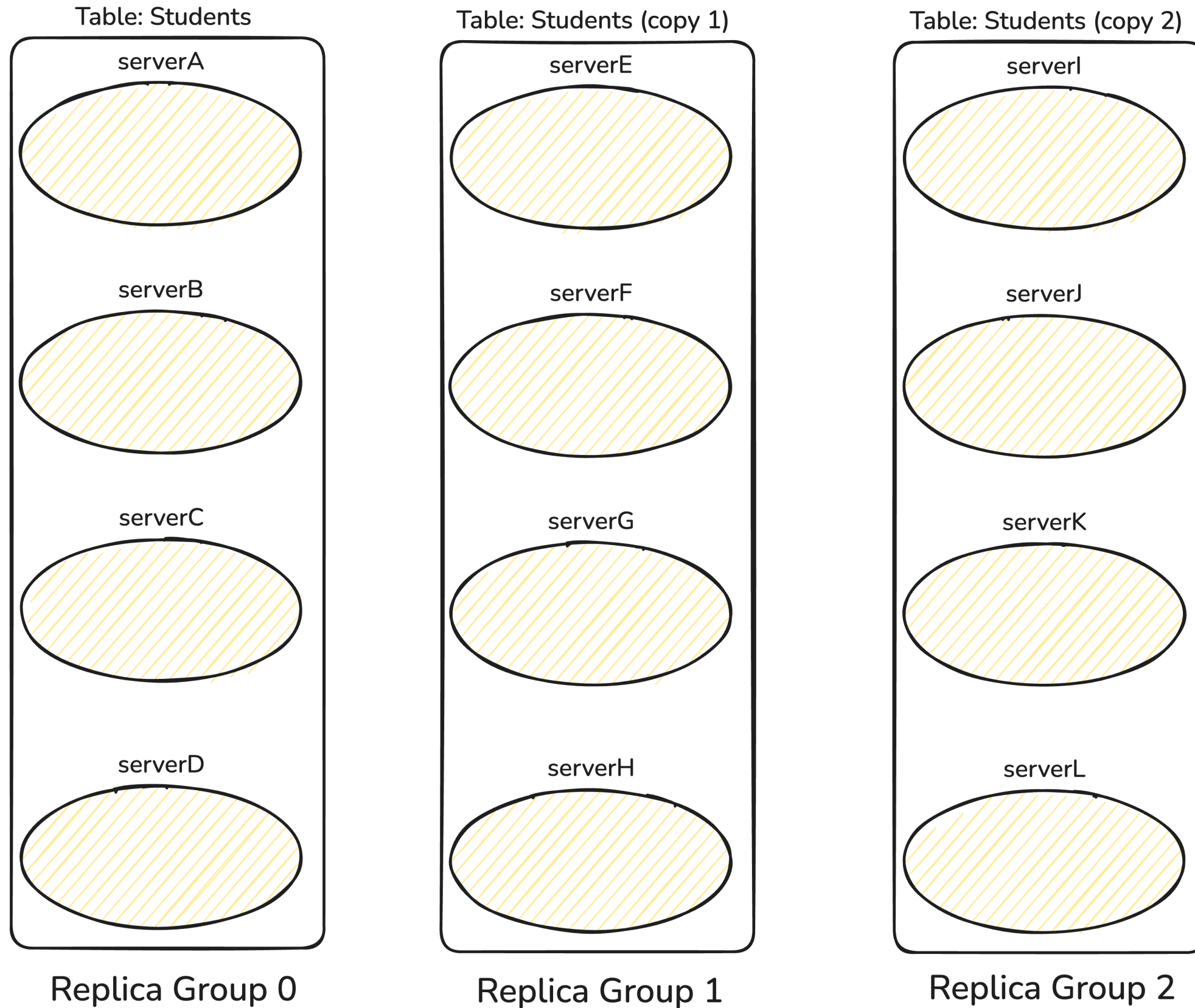
Shards



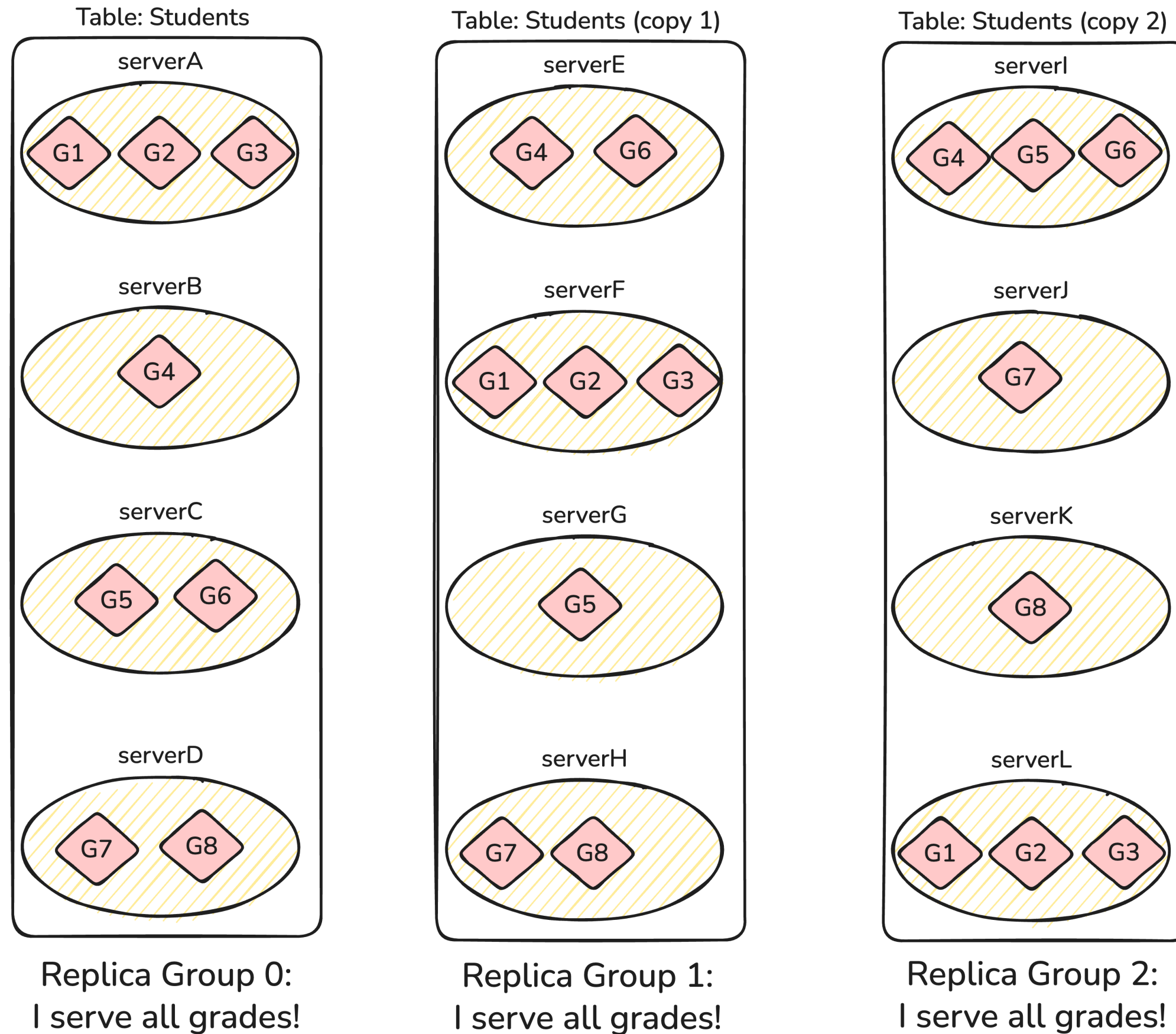
Replica Groups: How to place shards in hosts



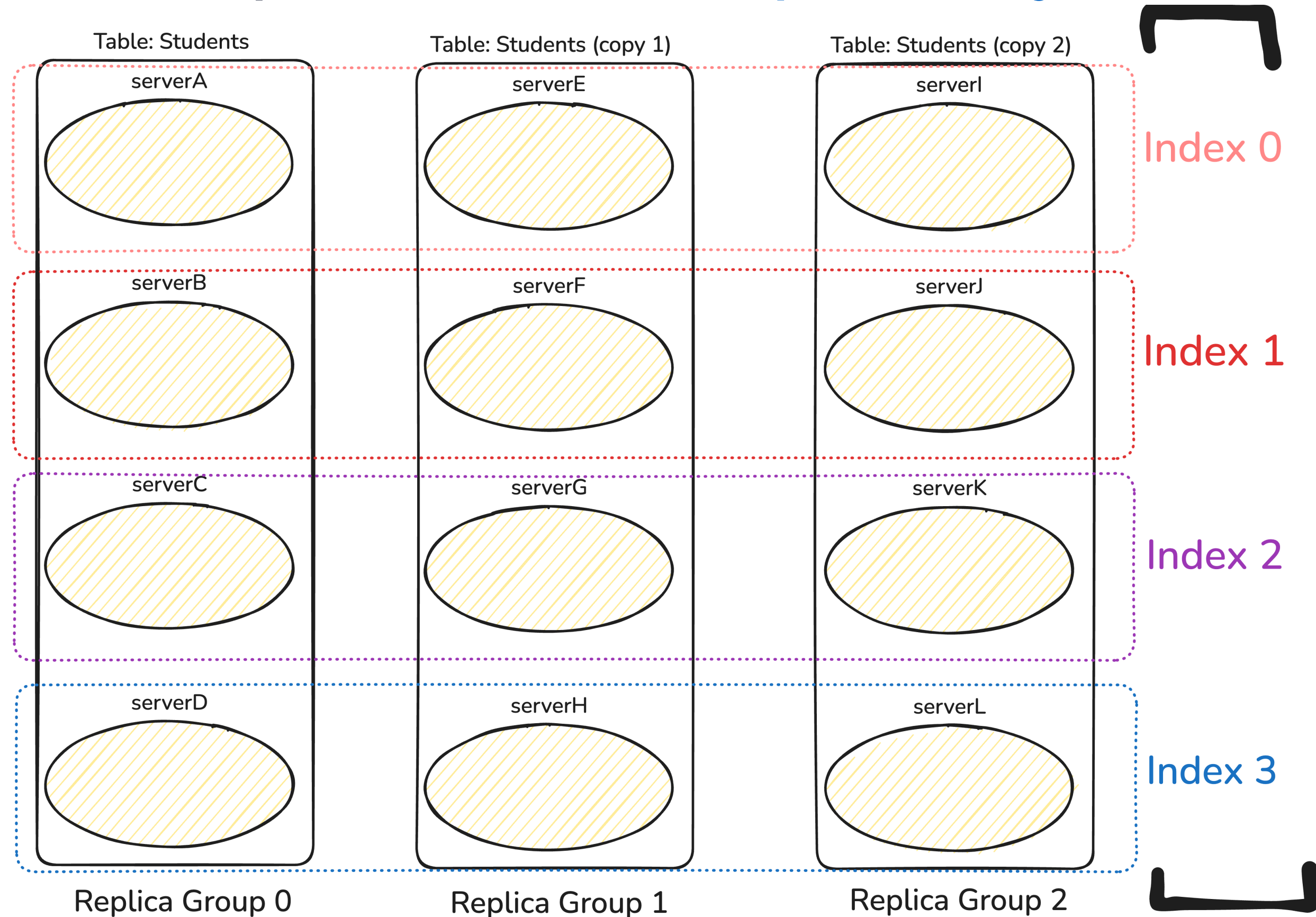
Replica Groups: How to place shards in hosts



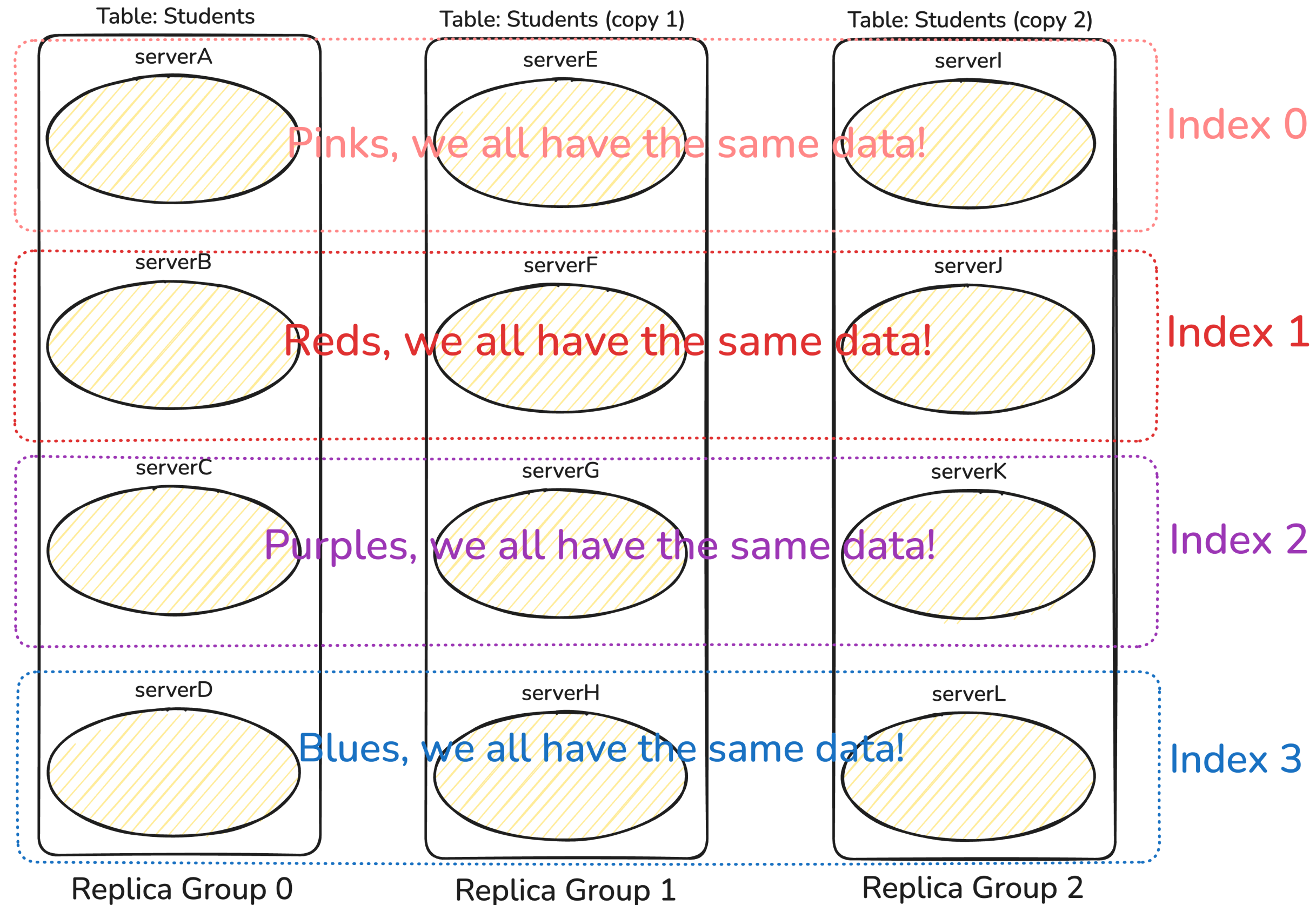
Replica Groups: How to place shards in hosts



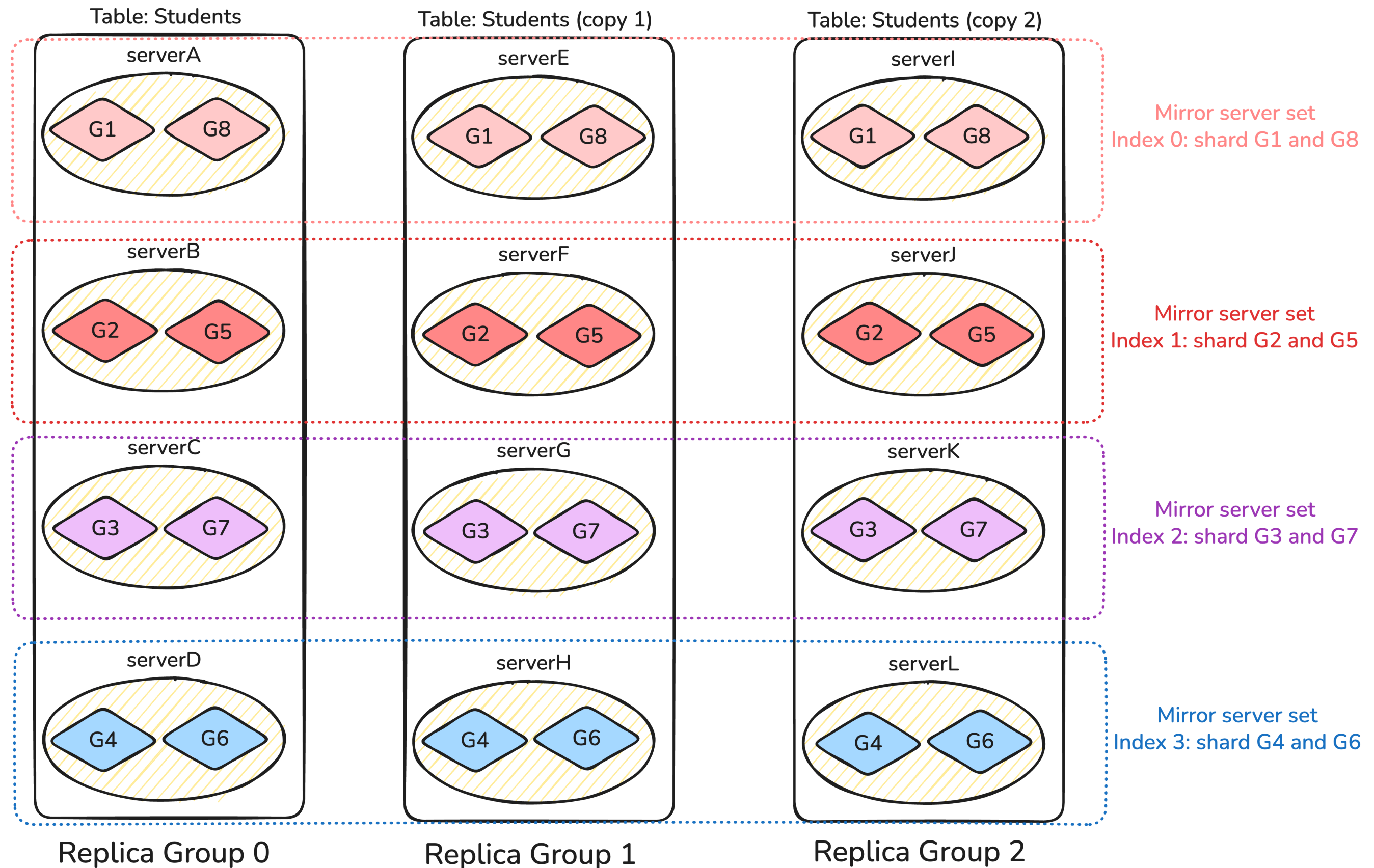
Mirror Server Set: How to place shards in hosts **predictably**



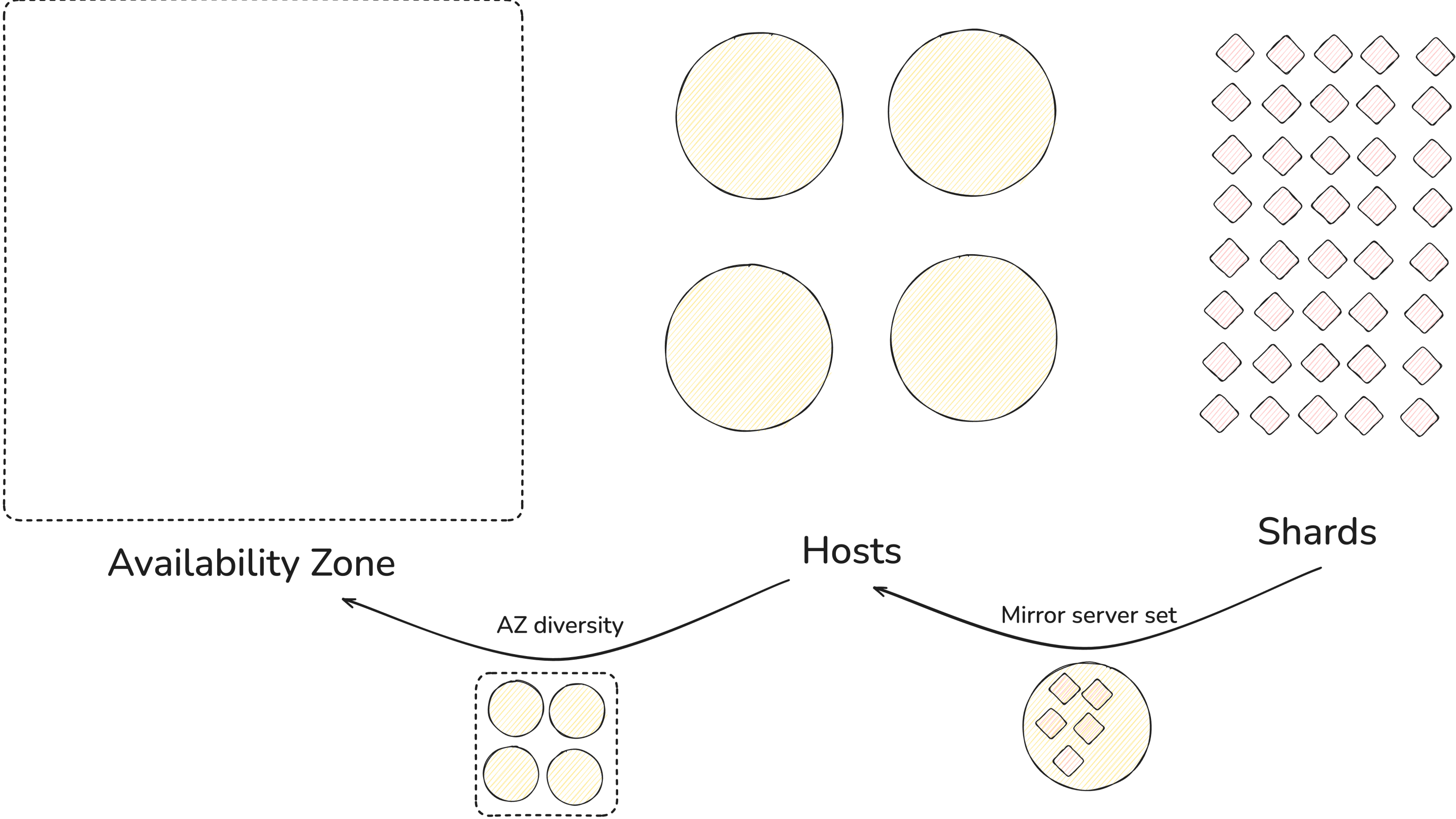
Mirror Server Set: How to place shards in hosts **predictably**



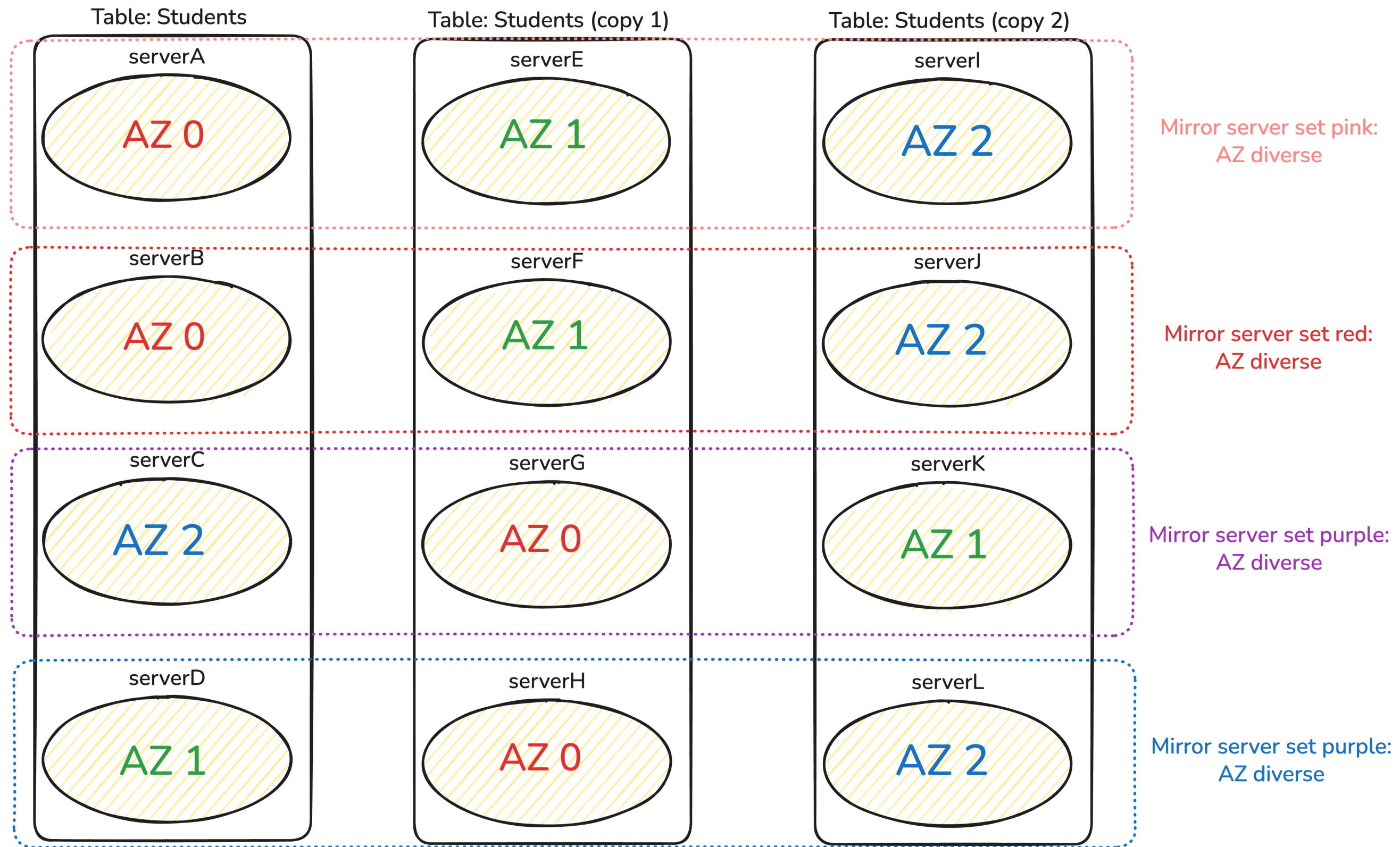
Mirror Server Set: How to place shards in hosts **predictably**



Adjustable Knobs for Shard Placement



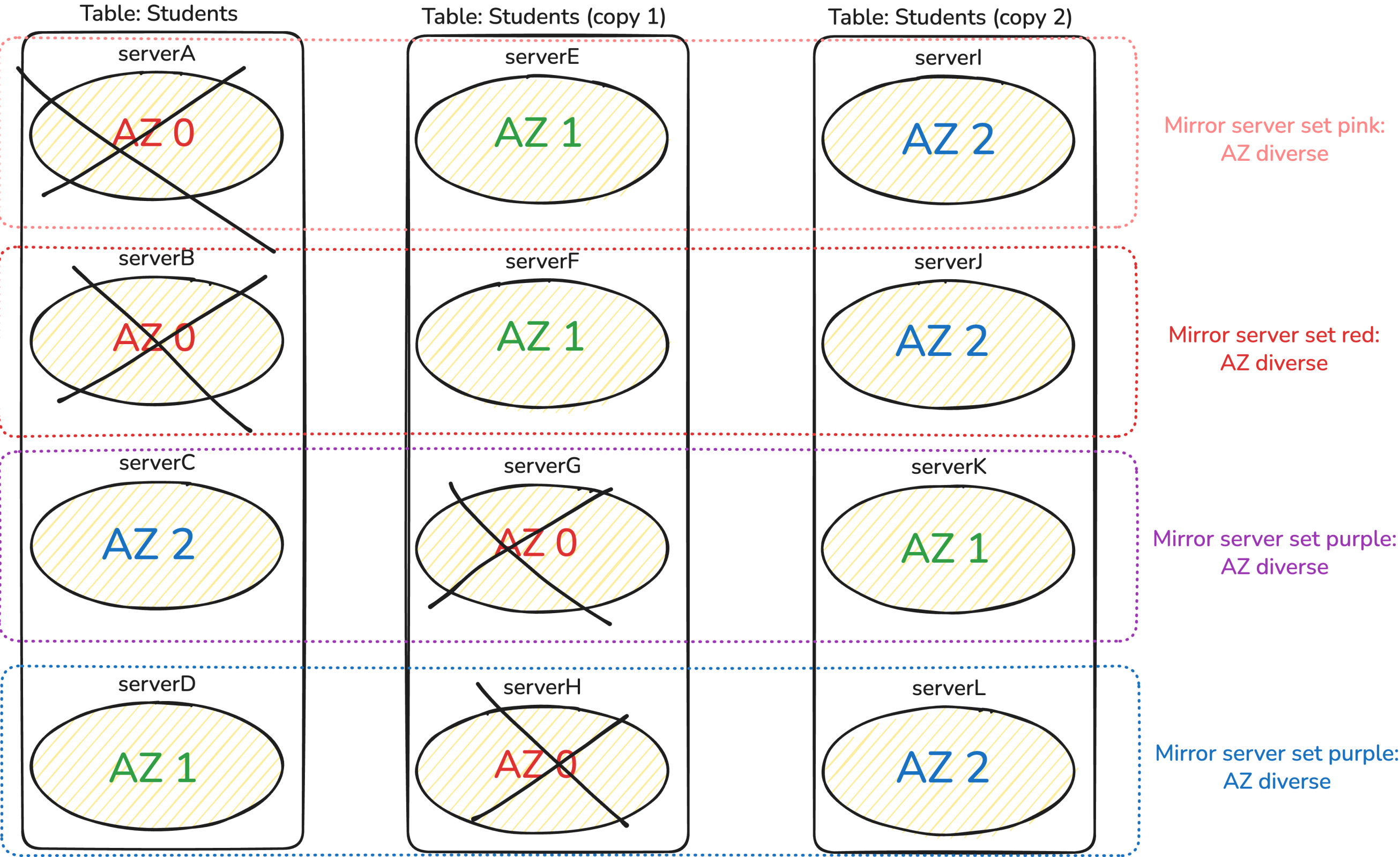
Mirror Server Set + Availability Zone Diversity



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

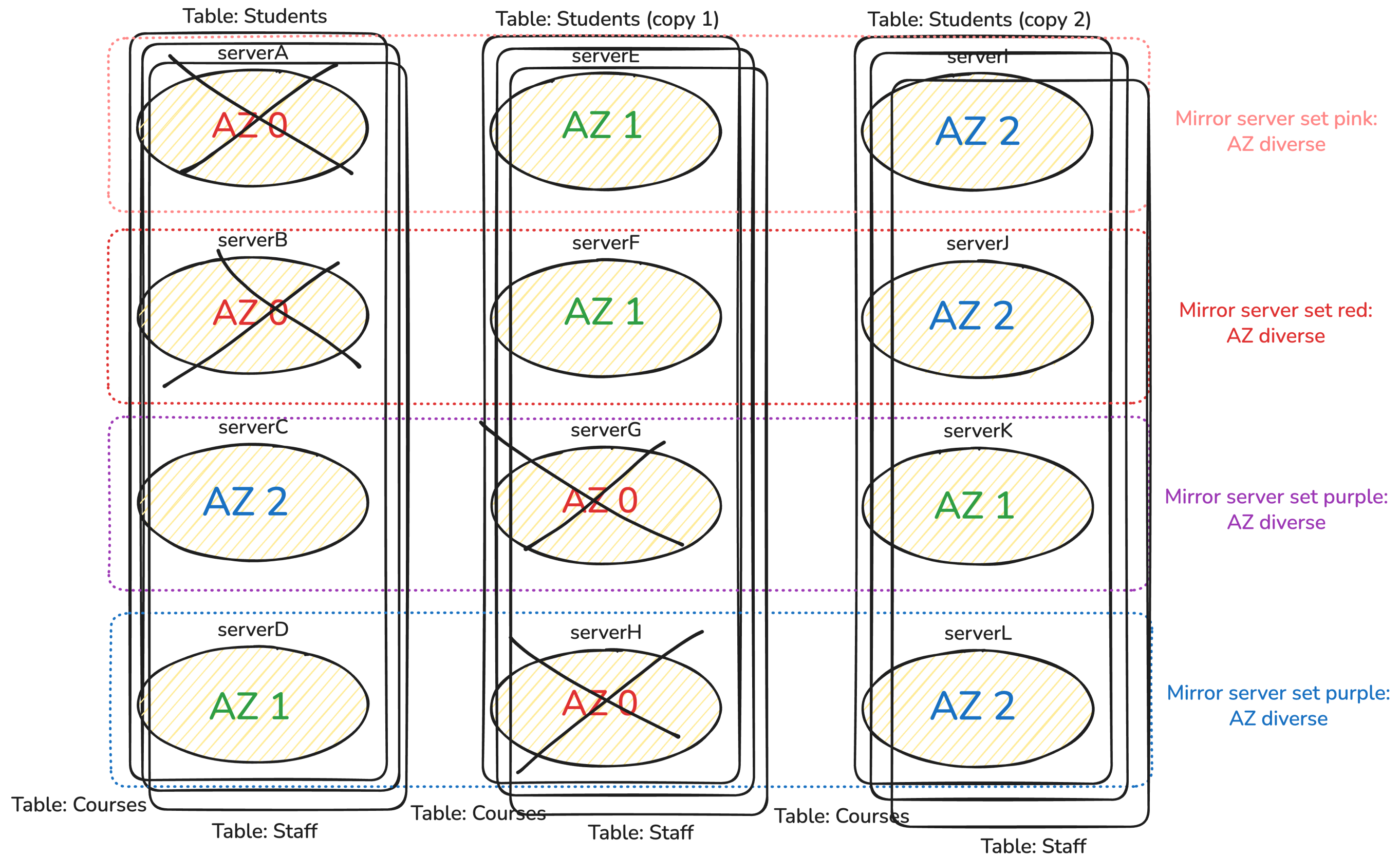


Mirror Server Set + Availability Zone Diversity



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

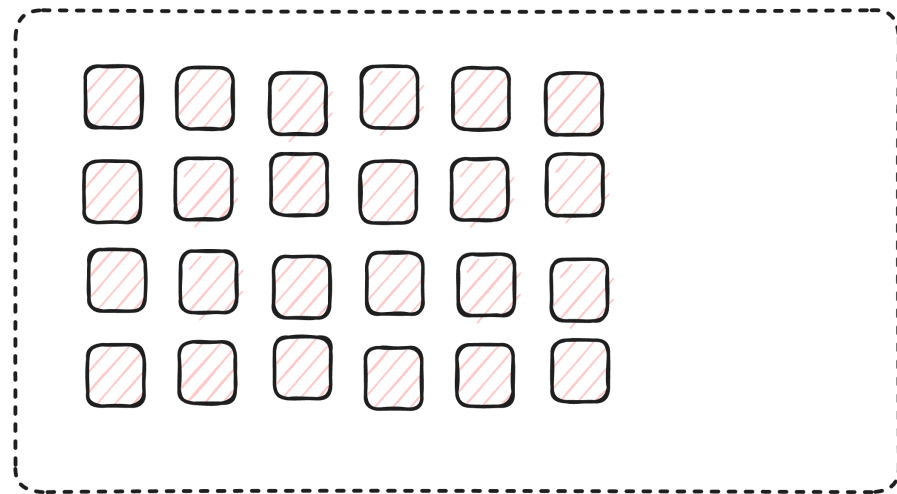


4

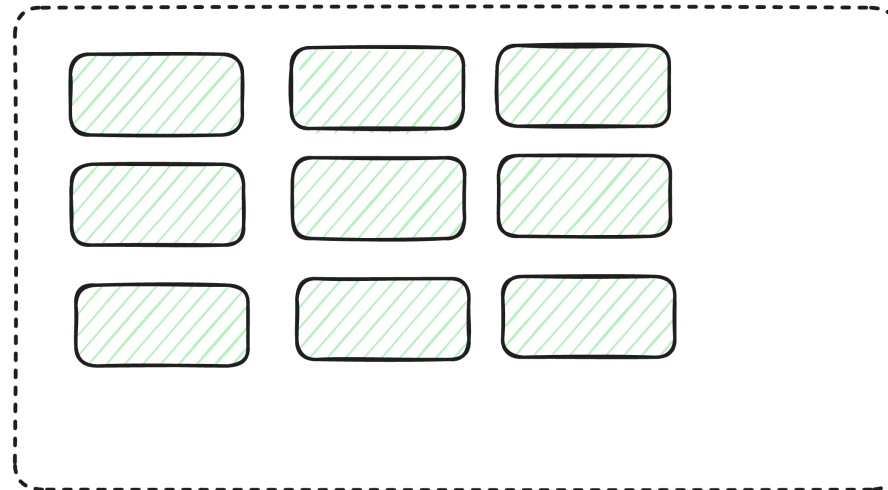
Multi-SKU Management

Manage node pools via k8s label:
pinot-server-*{RAM_SIZE}*-*{SSD_SIZE}*-SSD

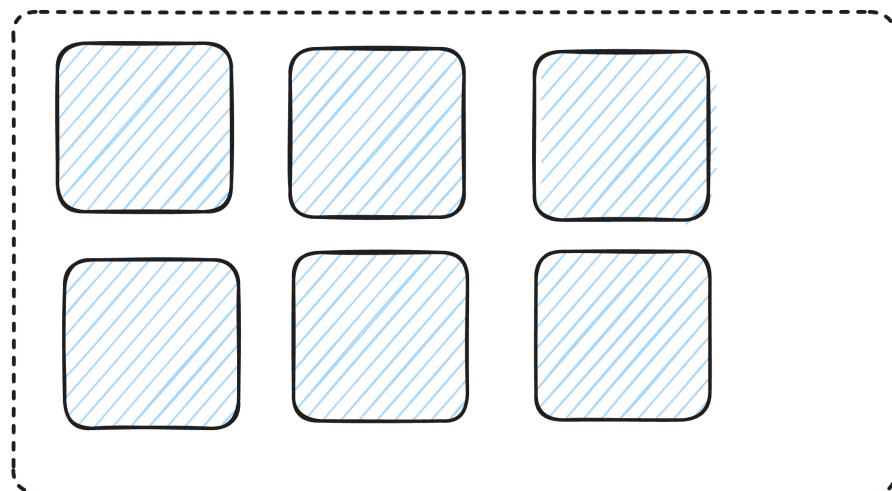
pinot-server-64GB-SSD



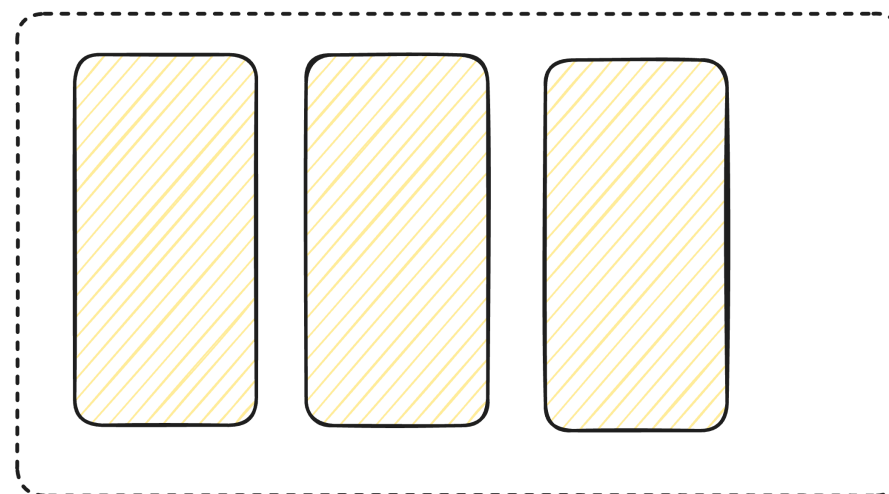
pinot-server-128GB-SSD



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

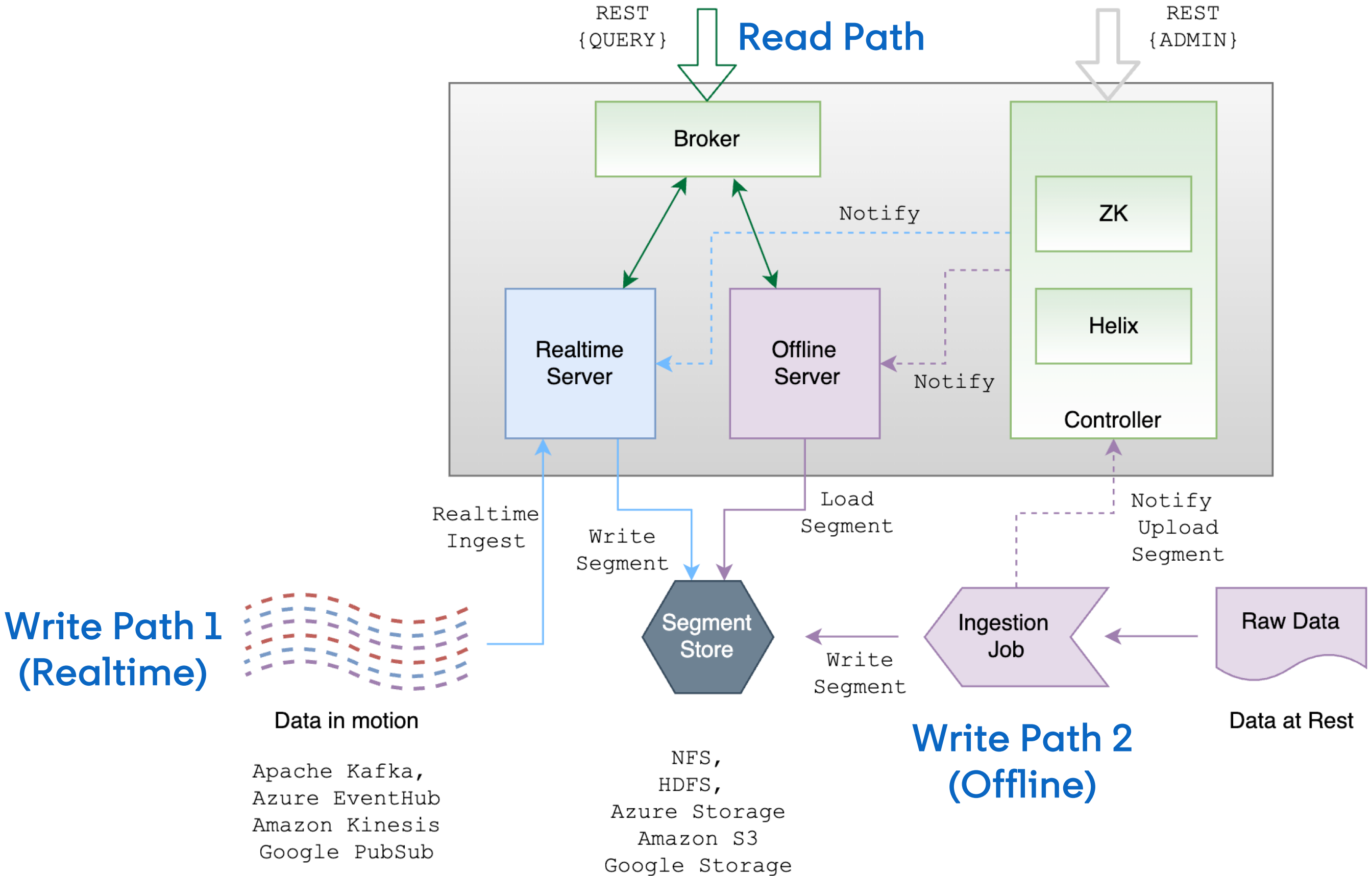
Pinot specific puppet modules

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

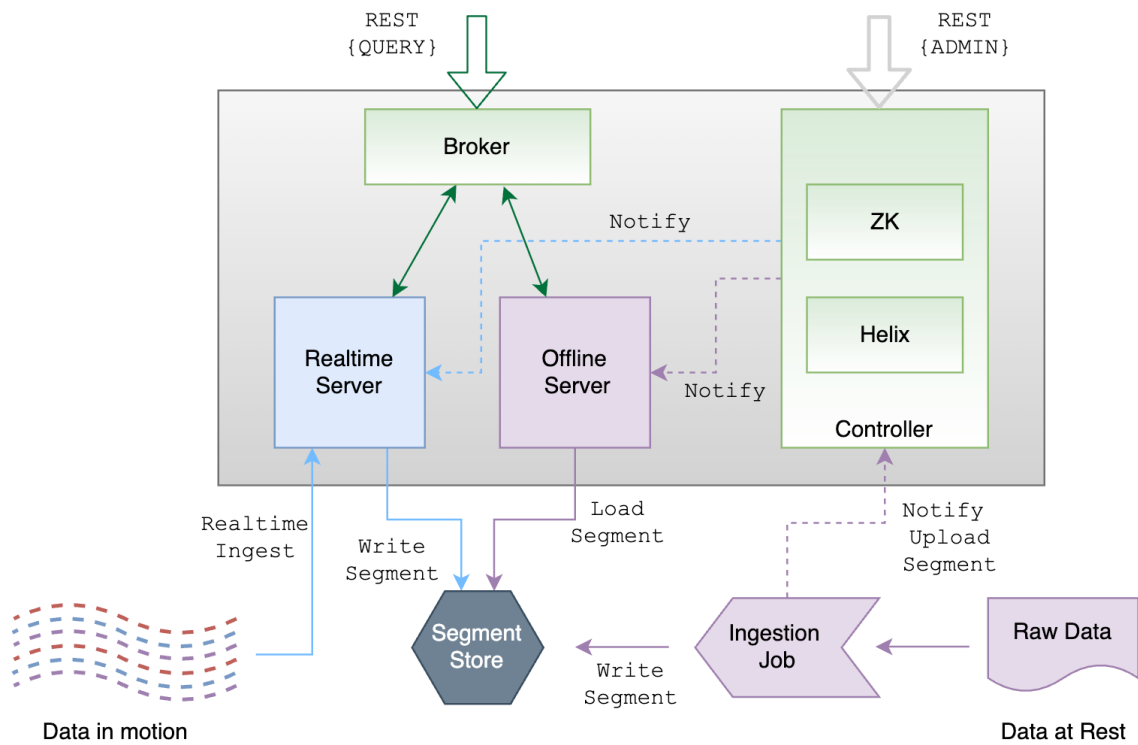
Alternatives

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

Stateful vs. Stateless Components



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

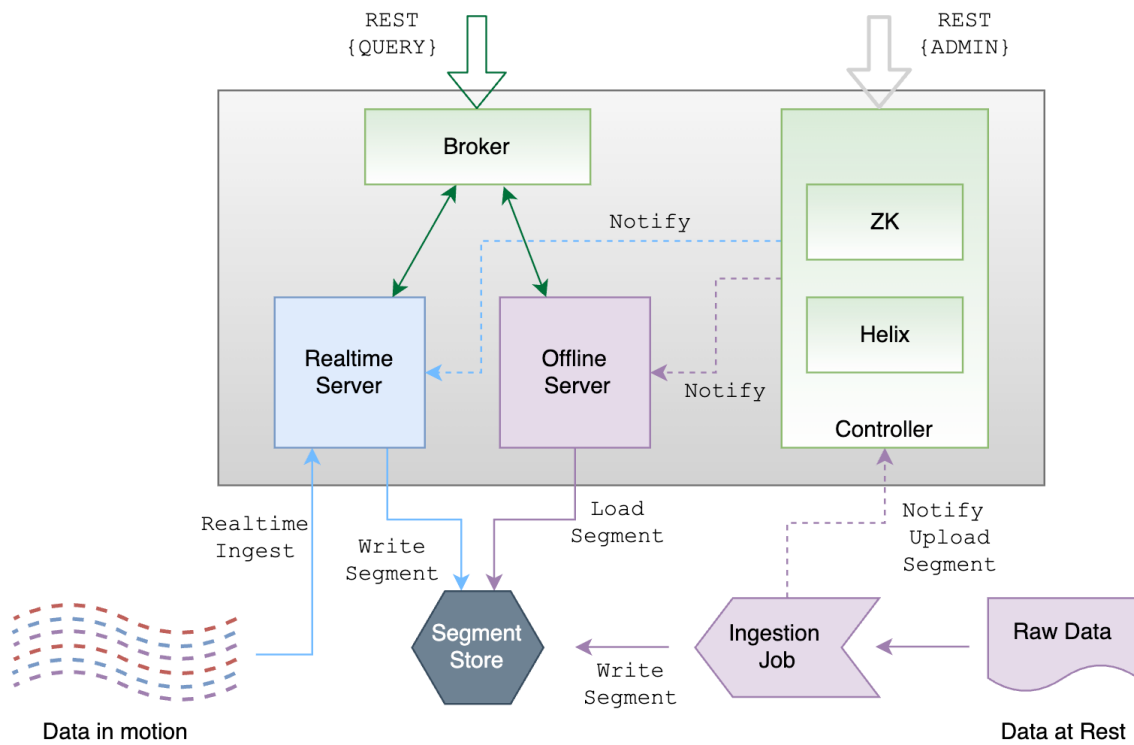
Persist data on host disk

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

Stateless Applications

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

Stateful Applications

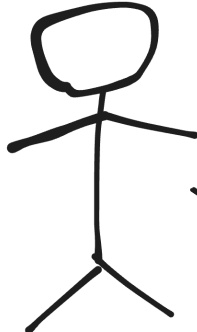
4 Offline server

Host shards and serve queries

Persist data on host disk

Host Maintenance – Application Cluster Manager (ACM)

Human/Automation



1. I want to stop serverA and serverB



2. Check the replica states for all shards served by servers A and B.
3. Verify that enough replicas are UP before stopping servers A and B.

4. Hmm..
You can only stop serverB.
Do NOT touch serverA please.



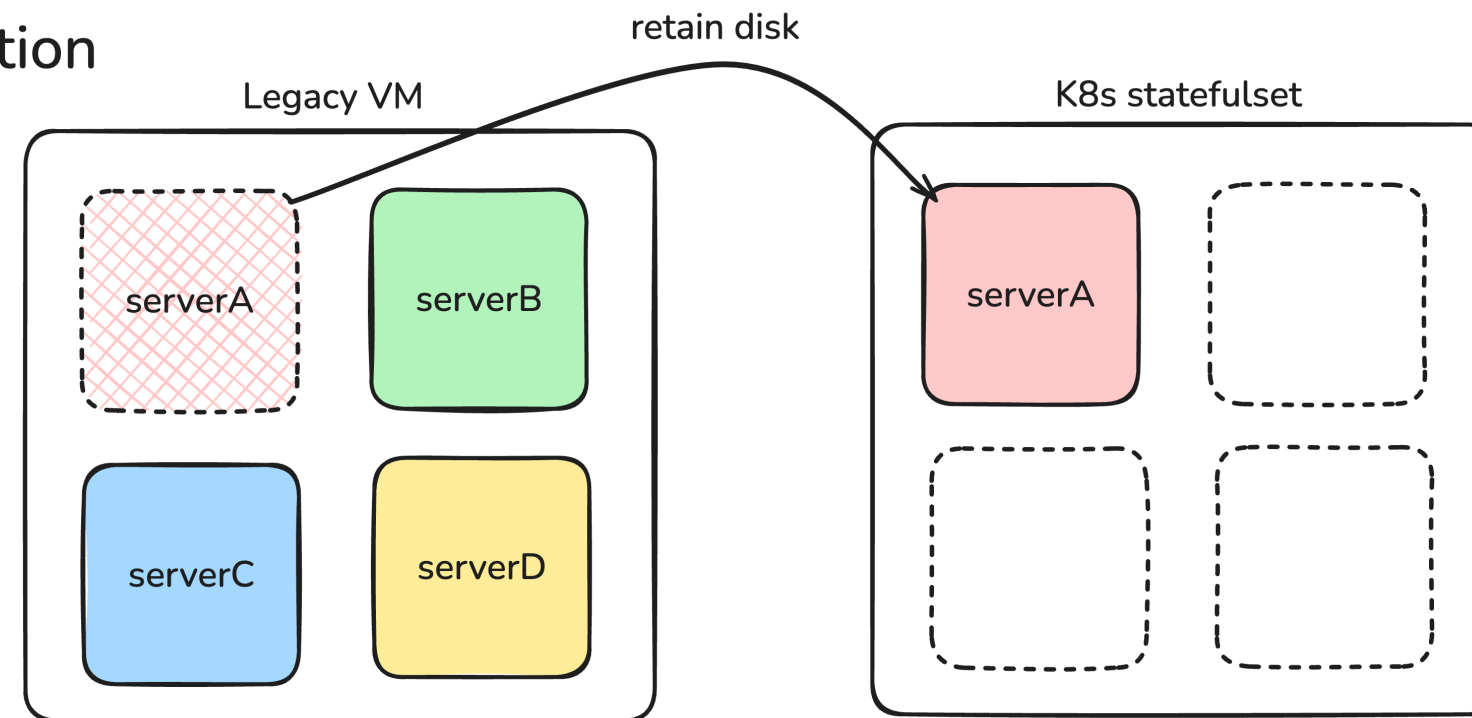
Migration Orchestrator: Design



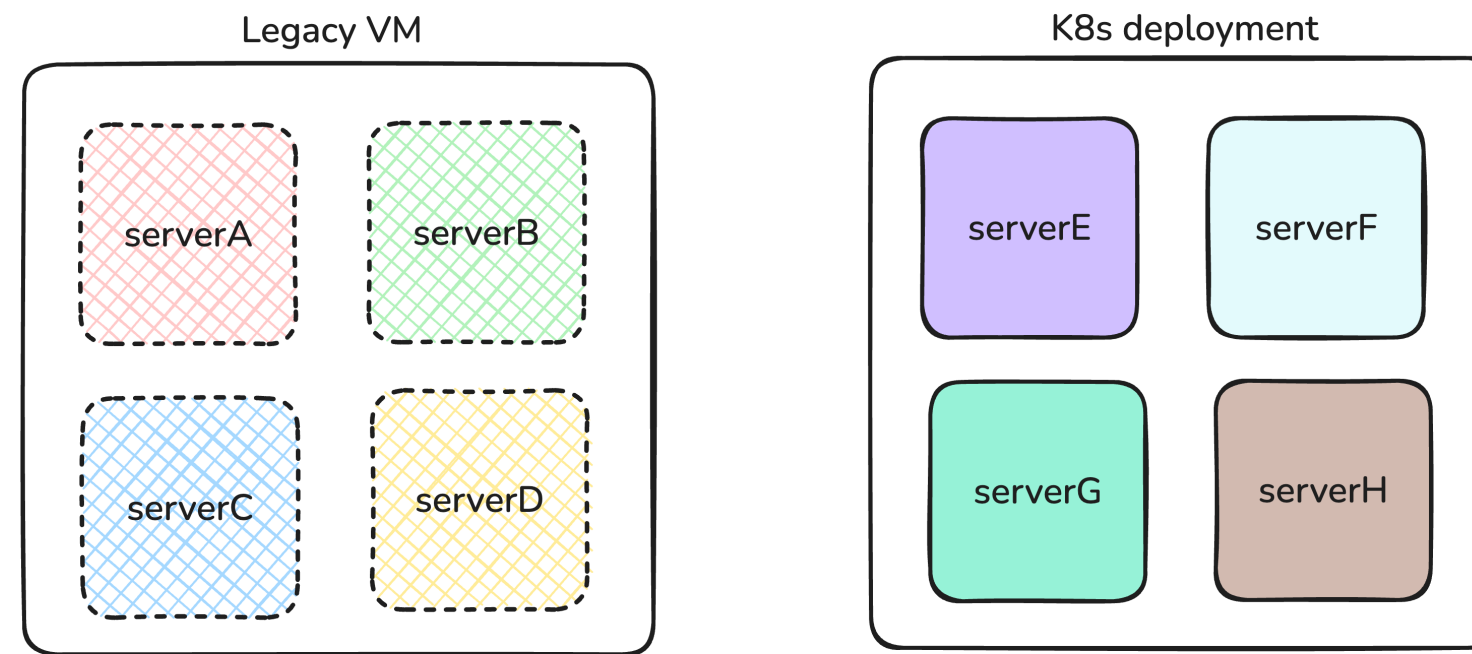
5

In-Place vs. Out-of-Place Migrations

In-Place Migration



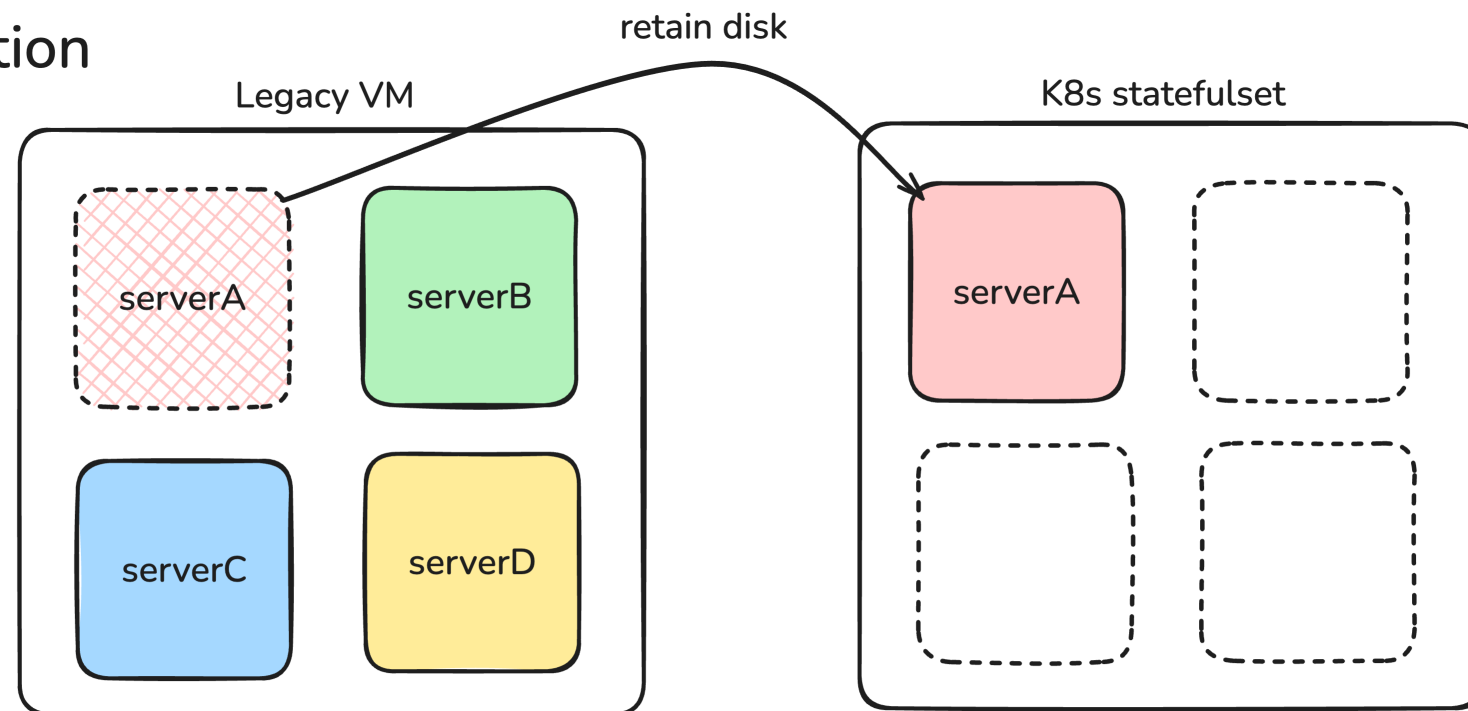
Out-of-Place Migration



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

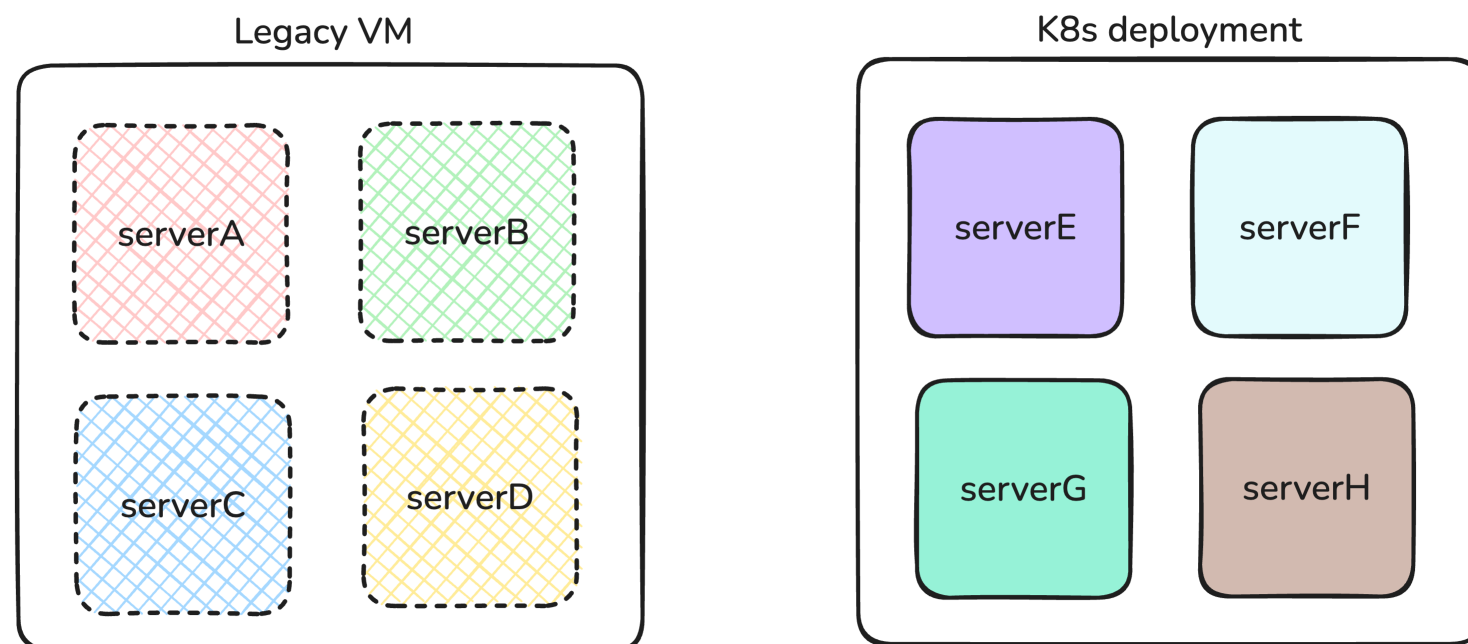
In-Place Migration



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

Stateful Applications

Out-of-Place Migration

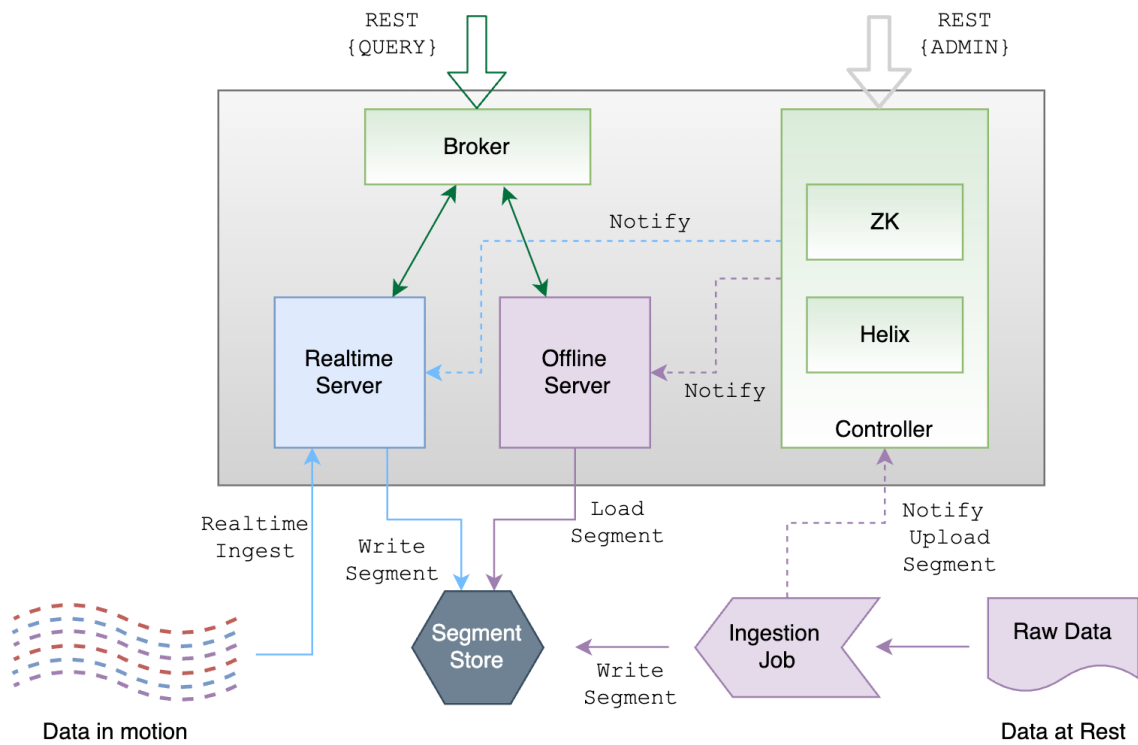


take down serverA, B, C, and D after the new set of servers are UP in the target K8s deployment

- + 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

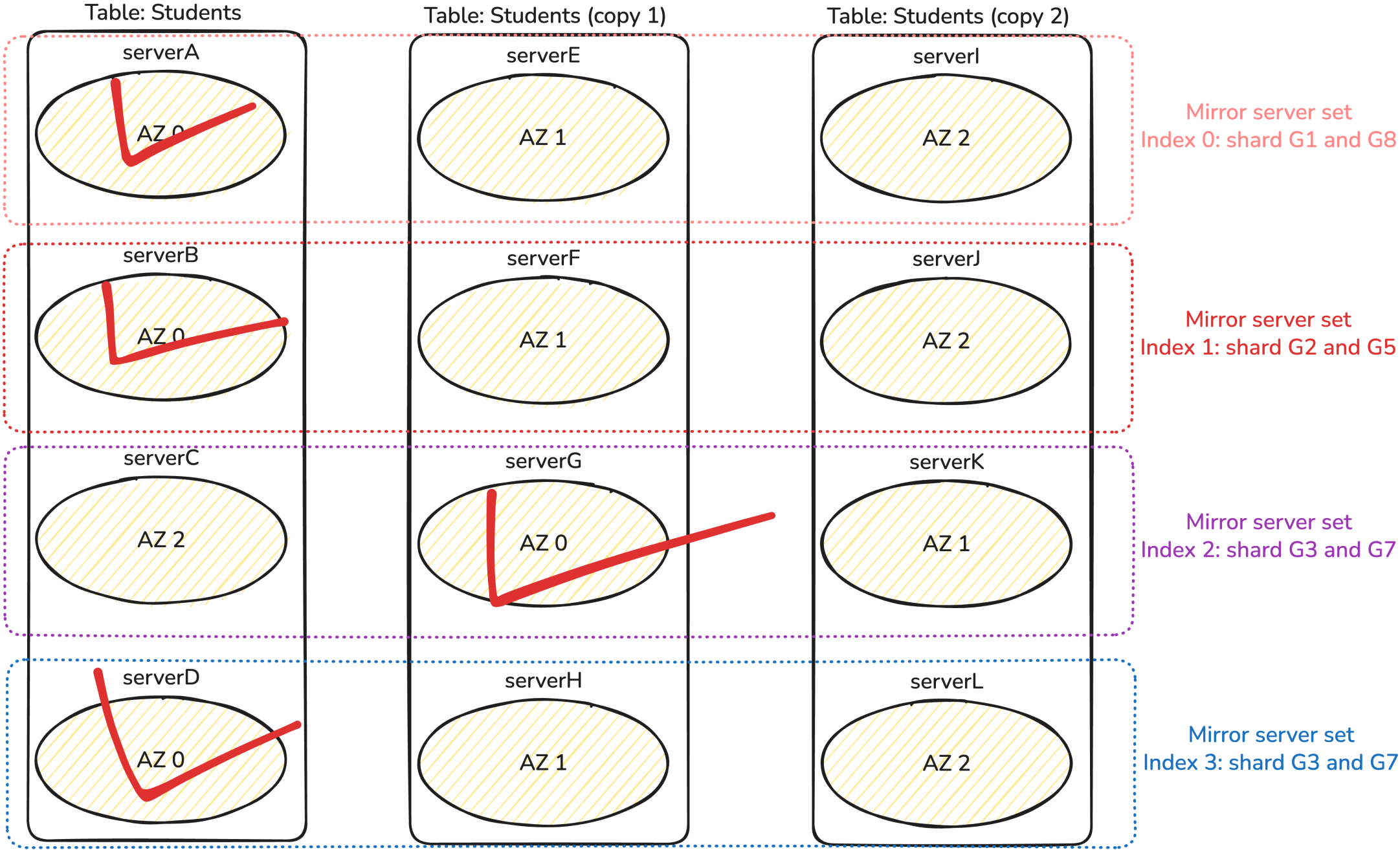
Stateful Applications
-> In-Place Migration

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

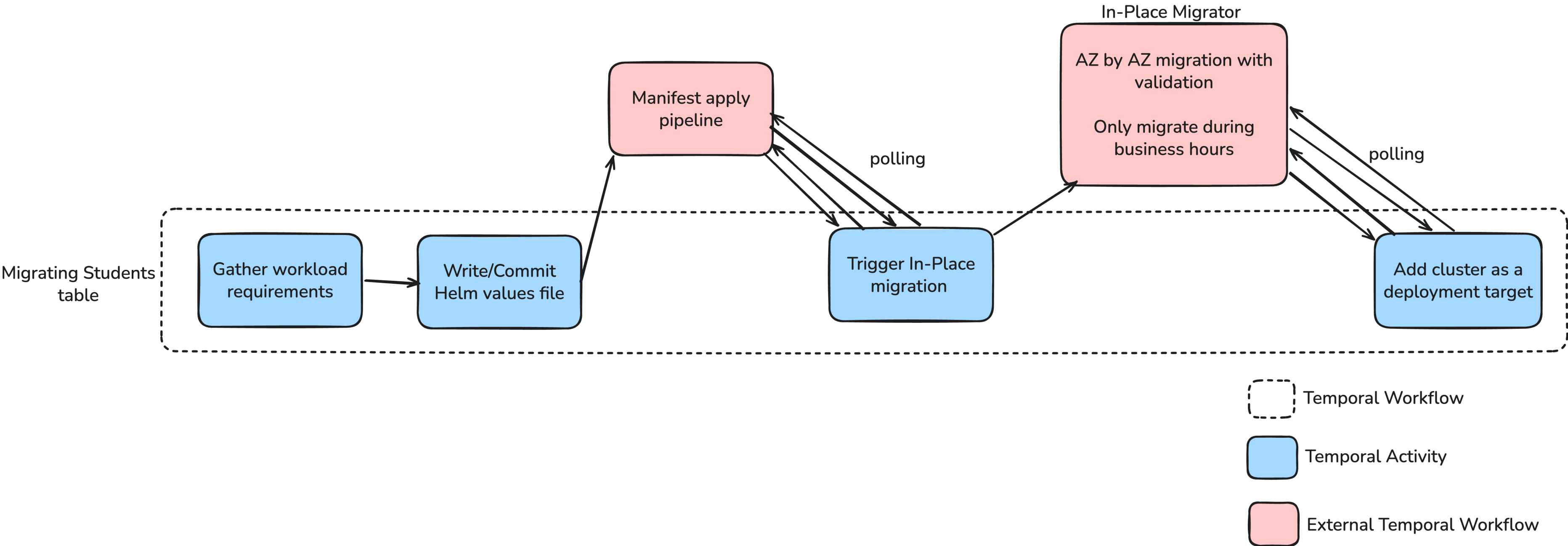
- Is the host healthy?
- Log validation
 - Are there any new errors?
 - Is the host ingesting data?
- Metric validation (compared to existing hosts)
 - Is the query latency in expected range?
 - Is CPU usage in expected range?

Migration Orchestrator: Execution

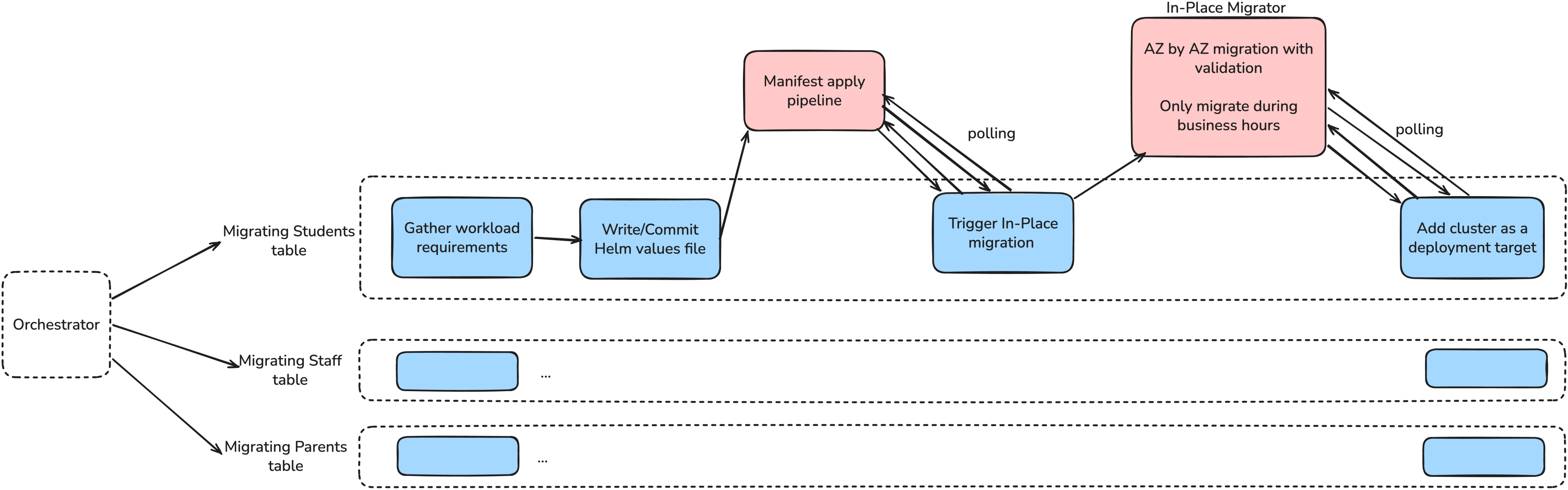


6

Temporal Workflow for In-Place Migration



Temporal Workflow for In-Place Migration



Temporal Workflow Used

Running prod-lva1--migration-2025-11-07T08:44:35Z Request Cancellation

🕒 3months 16d 23h 39m 2s 492... Start 2025-11-07 PST 00:44:35.79 Workflow Type Orchestration Task Queue pinot-temporal-prod-lva1
End - Run ID b774dd85-6c99-49c7-864a-04b67bb... History Size (Bytes) 305157

History 1462 Workers 0 **Relationships 215** Pending Activities 0 Call Stack Queries Metadata

Schedule

-migration

100

Status	Type	Child Workflow ID	Child Run ID
Completed	Tenant Migration	server--migration-b2f9f3e3-a1f4-4da3-9f6d-76ab99c2e9b8	624618e4-83b0-48e9-95a8-2b8c322e550d
Completed	Tenant Migration	server--migration-6e9a047b-a45b-4a4b-beef-c790dc201548	bba85595-02e7-47e9-a068-a66e07bc5e55
Completed	Tenant Migration	server--migration-a9eebab1-1bed-4df7-a215-43c71690ca94	793dc4d3-a9ed-4dda-8c2f-f940c3afc580
Completed	Tenant Migration	server--migration-1946b830-c829-4af1-9a4f-51e76ab1d979	2066eec5-27ae-4bb1-8fda-4084ffc5b641
Completed	Tenant Migration	server--migration-4f69e1d7-177f-4849-8288-9beb605395c5	f9ed7faa-853e-4c0d-b4de-a7fdb75d093
Completed	Tenant Migration	server--migration-804f899c-d9f2-4a8b-878a-0aa55130ff20	ebb2f636-1b0e-4002-96b8-a1170678e463
Completed	Tenant Migration	server--migration-ac2ef0ca-4153-45e3-a37a-9610b940cc1e	b5009dc5-581e-44ce-bef0-9f9bc3c5f845
Completed	Tenant Migration	server--migration-64af385c-f3bf-4b6d-a64f-9e77b73c3da9	cac21bd3-0f65-4c82-9c3f-c1aed0602b83
Completed	Tenant Migration	server--migration-b3072c90-44f7-4389-8591-7c2b7f75dc1e	ad8b1e98-2eef-47c9-bdac-2e2c45fee808
Completed	Tenant Migration	server--migration-d87bbf8d-898f-4aa8-b83f-0cfb32c7abd3	13d5fe27-6f96-4141-9843-5c1b87bdeedd8
Completed	Tenant Migration	server--migration-b339e198-ae75-4d83-9b63-643c8d5b8b29	894ed8ac-0a27-4eb7-822c-25896eead5d2
Completed	Tenant Migration	server--migration-5989403d-b99d-4aad-8c9c-6badf9b5dd55	8bcc5e21-2fd3-4994-b69c-06a3b29307e3
Completed	Tenant Migration	server--migration-b1b81867-ec72-4d0a-ad92-5c5b514938d1	a6f5cd6a-1334-443d-bf6f-a8ad3f0447e7



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/](https://www.linkedin.com/in/tony-song-dev/)

