



# Aurora extract:

DoorDash's journey from Aurora Postgres to CockroachDB

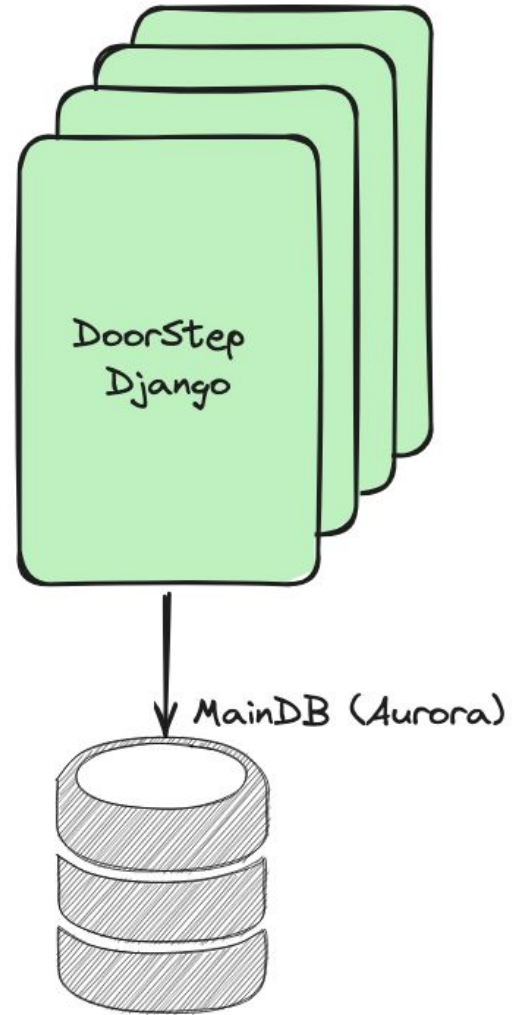
Alessandro Salvatori, Principal Engineer

**FRIDAY PARTY?**

**YES, SURE, I WILL JOIN!**









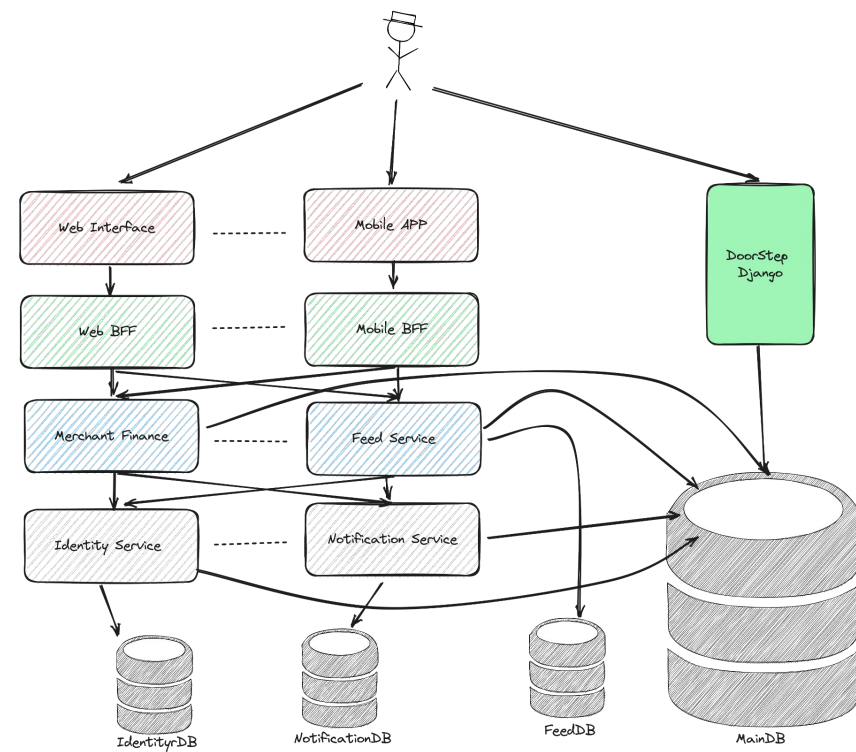


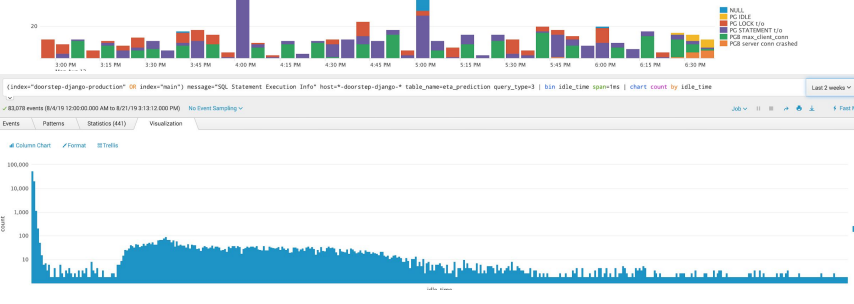
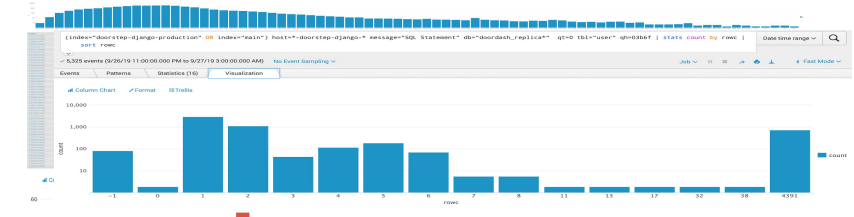
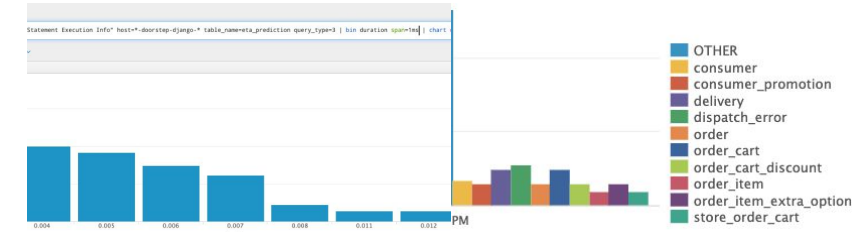
# SURVIVAL STRATEGY



Whack-a-mole

❖ Do not block microservices extractions





total idle time    timestamp    total duration    num. queries    diff. logs    sqlblock

679.898024    9:56:50:417477    756.897926    40    index=>doorstep-django-production\* host=>web\*    8b33ac0d-513f-442f-9f5c-0b497-

| id         | sum         | max       | min | avg | stddev | count |
|------------|-------------|-----------|-----|-----|--------|-------|
| 666.019678 | <-- sum --> | 82.948208 |     |     |        |       |
| 161.706209 | <-- max --> | 8.771896  |     |     |        |       |

| ms idle before | timestamp      | ms duration | rowcount | sql_type | table name  | sql                                       |
|----------------|----------------|-------------|----------|----------|---|---|
| 53.452015      | 9:56:49:716298 | 1.918793    | 1        | 0        | order_cart  | SELECT ... FROM "order_cart" WHERE `      |
| 4.360199       | 9:56:49:721825 | 1.291990    | 1        | 0        | consumer  | SELECT ... FROM "consumer" WHERE `        |
| 12.748003      | 9:56:49:735918 | 1.359940    | 1        | 0        | order_cart  | SELECT ... FROM "order_cart" WHERE `      |
| 2.929266       | 9:56:49:740157 | 1.301050    | 1        | 0        | consumer  | SELECT ... FROM "consumer" WHERE `        |
| 10.919094      | 9:56:49:759895 | 8.771896    | 1        | 3        | delivery  | INSERT INTO "delivery" ... RETURNING `    |
| 11.534929      | 9:56:49:773329 | 1.899004    | 1        | 0        | delivery  | SELECT ... FROM "delivery" WHERE `del     |
| 51.705122      | 9:56:49:826172 | 1.255305    | 1        | 0        | delivery_assignment_constraint                          | SELECT ... FROM "delivery_assignment_co   |
| 17.097950      | 9:56:49:850492 | 7.222891    | 1        | 3        | drive_delivery_identifier_mapping                       | INSERT INTO "drive_delivery_identifier_m  |
| 3.581047       | 9:56:49:855345 | 1.235962    | 1        | 3        | delivery_item   | INSERT INTO "delivery_item" ... RETURN    |
| 2.228975       | 9:56:49:858587 | 1.129150    | 1        | 3        | delivery_item   | INSERT INTO "delivery_item" ... RETURN    |
| 6.001949       | 9:56:49:866100 | 1.501083    | 0        | 0        | drive_order   | SELECT ... FROM "drive_order" WHERE `     |
| 2.973795       | 9:56:49:870489 | 1.418114    | 0        | 0        | drive_order   | INSERT INTO "drive_order" WHERE `         |
| 2.604961       | 9:56:49:874160 | 1.071930    | 0        | 0        | delivery_receipt  | SELECT ... FROM "delivery_receipt" WHE    |
| 3.015995       | 9:56:49:879025 | 1.718044    | 1        | 3        | delivery_drive_info                                     | INSERT INTO "delivery_drive_info" ("creat |
| 8.240938       | 9:56:49:888277 | 1.133204    | 1        | 0        | order   | SELECT ... FROM "order" WHERE ("order     |
| 2.528906       | 9:56:49:891848 | 1.015902    | 1        | 0        | order_item  | SELECT COUNT(*) AS `__count` FROM `       |
| 161.706209     | 9:56:50:055231 | 1.552820    | 1        | 3        | eta_prediction  | INSERT INTO "eta_prediction" ... RETURN   |
| 4.763126       | 9:56:50:062655 | 2.791882    | 1        | 0        | delivery  | UPDATE "delivery" SET "active_date" = `   |
| 7.066011       | 9:56:50:071631 | 1.914024    | 1        | 0        | delivery  | SELECT ... FROM "delivery" WHERE `del     |
| 53.141117      | 9:56:50:126207 | 1.366881    | 1        | 0        | eta_prediction  | UPDATE "eta_prediction" SET ` WHERE       |
| 3.975153       | 9:56:50:131863 | 1.621962    | 1        | 0        | delivery  | UPDATE "delivery" SET "active_date" = `   |
| 6.289005       | 9:56:50:139977 | 1.924992    | 1        | 0        | delivery  | SELECT ... FROM "delivery" WHERE `del     |
| 40.563822      | 9:56:50:182858 | 2.307177    | 0        | 0        | delivery_event  | SELECT ... FROM "delivery_event" WHEF     |
| 5.472898       | 9:56:50:190311 | 1.781940    | 1        | 3        | delivery_event  | INSERT INTO "delivery_event" ... RETUR    |
| 6.786027       | 9:56:50:198315 | 1.336098    | 2        | 0        | delivery_event_category                                 | SELECT ... FROM "delivery_event_categ     |
| 12.544870      | 9:56:50:212795 | 1.912117    | 1        | 0        | delivery  | SELECT ... FROM "delivery" WHERE `de      |
| 13.639927      | 9:56:50:232299 | 6.814075    | 1        | 0        | eta_prediction  | SELECT ... FROM "eta_prediction" WHE      |
| 41.621923      | 9:56:50:276743 | 2.799034    | 1        | 0        | store_order_cart  | SELECT ... FROM "store_order_cart" WHI    |
| 15.609202      | 9:56:50:293864 | 1.584053    | 1        | 0        | order_item  | SELECT ... FROM "order" UO WHERE UO       |
| 19.042969      | 9:56:50:314141 | 1.211166    | 1        | 0        | order_cart_pricing_strategy                             | SELECT ... FROM "order_cart_pricing_st    |
| 5.800099       | 9:56:50:321555 | 1.612902    | 0        | 0        | consumer_promotion & order_cart_consumer_promotion_link | SELECT (1) AS "a" FROM "consumer_prc      |
| 4.657090       | 9:56:50:329769 | 1.780987    | 0        | 0        | consumer_promotion & order_cart_consumer_promotion_link | SELECT ... FROM "consumer_promotion"      |
| 7.226944       | 9:56:50:340162 | 3.051043    | 0        | 0        | delivery  | SELECT ... FROM "delivery" WHERE `de      |
| 8.162022       | 9:56:50:350101 | 1.968953    | 1        | 0        | delivery  | SELECT ... FROM "delivery" WHERE `de      |
| 5.224943       | 9:56:50:356722 | 1.496077    | 1        | 1        | store_order_cart  | UPDATE "store_order_cart" SET ` WHEN      |
| 8.534908       | 9:56:50:366741 | 1.469135    | 1        | 1        | order_cart  | UPDATE "order_cart" SET "submitted_at"    |
| 3.969908       | 9:56:50:372360 | 1.605304    | 1        | 0        | delivery  | SELECT ... FROM "delivery" WHERE `de      |
| 5.981922       | 9:56:50:379604 | 1.348972    | 1        | 0        | order_cart_discount                                     | SELECT ... FROM "order_cart_discount"     |
| 3.010988       | 9:56:50:384869 | 2.254409    | 1        | 1        | order_cart_discount                                     | UPDATE "order_cart_discount" SET "curr    |
| 23.573160      | 9:56:50:408623 | 1.160887    | 1        | 0        | order_cart  | SELECT (1) AS "a" FROM "order_cart" WI    |
| 13.878345      | 9:56:50:417477 | 7.930040    |          |          |   | COMMIT                                    |

./jq

| col | ops  | to/s   | read % | tbl     | 5784.56                            |
|-----|------|--------|--------|---------|------------------------------------|
| 0   | 3842 | 426.89 | 427.33 | 99.90%  | dispatch_error                     |
| 1   | 4    | 0.44   | 427.33 | 99.90%  | dispatch_error                     |
| 1   | 116  | 12.89  | 412.89 | 96.88%  | order                              |
| 0   | 3600 | 400.00 | 412.89 | 96.88%  | order                              |
| 0   | 2561 | 284.56 | 345.89 | 82.27%  | consumer                           |
| 1   | 552  | 61.33  | 345.89 | 82.27%  | consumer                           |
| 0   | 3052 | 339.11 | 339.11 | 100.00% | globalvars_variable                |
| 0   | 1542 | 171.33 | 309.11 | 55.43%  | order_cart                         |
| 1   | 1240 | 137.78 | 309.11 | 55.43%  | order_cart                         |
| 0   | 2382 | 264.67 | 264.78 | 99.96%  | consumer_promotion                 |
| 1   | 1    | 0.11   | 264.78 | 99.96%  | consumer_promotion                 |
| 0   | 945  | 105.00 | 258.67 | 40.59%  | store_order_cart                   |
| 1   | 1383 | 153.67 | 258.67 | 40.59%  | store_order_cart                   |
| 0   | 694  | 77.11  | 255.67 | 30.16%  | delivery_event                     |
| 1   | 1607 | 178.56 | 255.67 | 30.16%  | delivery_event                     |
| 0   | 1758 | 195.33 | 247.67 | 78.87%  | order_item                         |
| 1   | 471  | 52.33  | 247.67 | 78.87%  | order_item                         |
| 1   | 1540 | 171.11 | 224.33 | 23.72%  | order_cart_discount                |
| 0   | 479  | 53.22  | 224.33 | 23.72%  | order_cart_discount                |
| 0   | 959  | 106.56 | 210.33 | 50.66%  | delivery                           |
| 1   | 934  | 103.78 | 210.33 | 50.66%  | delivery                           |
| 0   | 1587 | 176.33 | 200.33 | 97.78%  | user_social_data                   |
| 1   | 36   | 4.00   | 180.33 | 97.78%  | user_social_data                   |
| 0   | 1265 | 140.56 | 158.11 | 88.90%  | order_item_extra_option            |
| 1   | 158  | 17.56  | 158.11 | 88.90%  | order_item_extra_option            |
| 1   | 10   | 1.11   | 151.00 | 99.26%  | stripe_card                        |
| 0   | 1349 | 149.89 | 151.00 | 99.26%  | stripe_card                        |
| 0   | 1121 | 124.56 | 131.11 | 95.00%  | stripe_charge                      |
| 1   | 59   | 6.56   | 131.11 | 95.00%  | stripe_charge                      |
| 0   | 1175 | 130.56 | 130.56 | 100.00% | promo_code                         |
| 0   | 1115 | 123.89 | 123.89 | 100.00% | country                            |
| 0   | 1025 | 113.89 | 114.22 | 99.71%  | delivery_drive_info                |
| 1   | 3    | 0.33   | 114.22 | 99.71%  | delivery_drive_info                |
| 1   | 55   | 6.11   | 97.44  | 93.73%  | delivery_issue                     |
| 0   | 822  | 91.33  | 97.44  | 93.73%  | delivery_issue                     |
| 0   | 780  | 86.67  | 86.67  | 100.00% | marqeta_card_ownership             |
| 0   | 648  | 72.00  | 84.11  | 85.60%  | order_cart_discount_component      |
| 1   | 109  | 12.11  | 84.11  | 85.60%  | order_cart_discount_component      |
| 0   | 739  | 82.11  | 82.11  | 100.00% | auth_group                         |
| 0   | 700  | 77.78  | 77.78  | 100.00% | order_cart_pricing_strategy        |
| 0   | 610  | 67.78  | 73.11  | 92.71%  | consumer_charge                    |
| 1   | 48   | 5.33   | 73.11  | 92.71%  | consumer_charge                    |
| 0   | 576  | 64.00  | 65.89  | 97.13%  | order_cart_consumer_promotion_link |
| 1   | 17   | 1.89   | 65.89  | 97.13%  | order_cart_consumer_promotion_link |
| 1   | 1    | 0.11   | 60.89  | 99.82%  | consumer_referral_link             |
| 0   | 547  | 60.78  | 60.89  | 99.82%  | consumer_referral_link             |
| 0   | 546  | 60.67  | 60.67  | 100.00% | transfer                           |
| 0   | 473  | 52.56  | 54.00  | 97.33%  | promotion_consumer_link            |
| 1   | 13   | 1.44   | 54.00  | 97.33%  | promotion_consumer_link            |
| 0   | 260  | 28.89  | 42.11  | 31.40%  | user                               |
| 0   | 119  | 13.22  | 42.11  | 31.40%  | user                               |
| 0   | 353  | 39.22  | 39.33  | 99.72%  | marqeta_card                       |
| 1   | 1    | 0.11   | 39.33  | 99.72%  | marqeta_card                       |
| 0   | 346  | 38.44  | 38.56  | 99.71%  | payment_account                    |
| 1   | 1    | 0.11   | 38.56  | 99.71%  | payment_account                    |
| 0   | 285  | 31.67  | 31.67  | 100.00% | delivery_event_category            |

SINGLE PRIMARY





MULTIPLE WRITERS





# SURVIVAL STRATEGY



Whack-a-mole

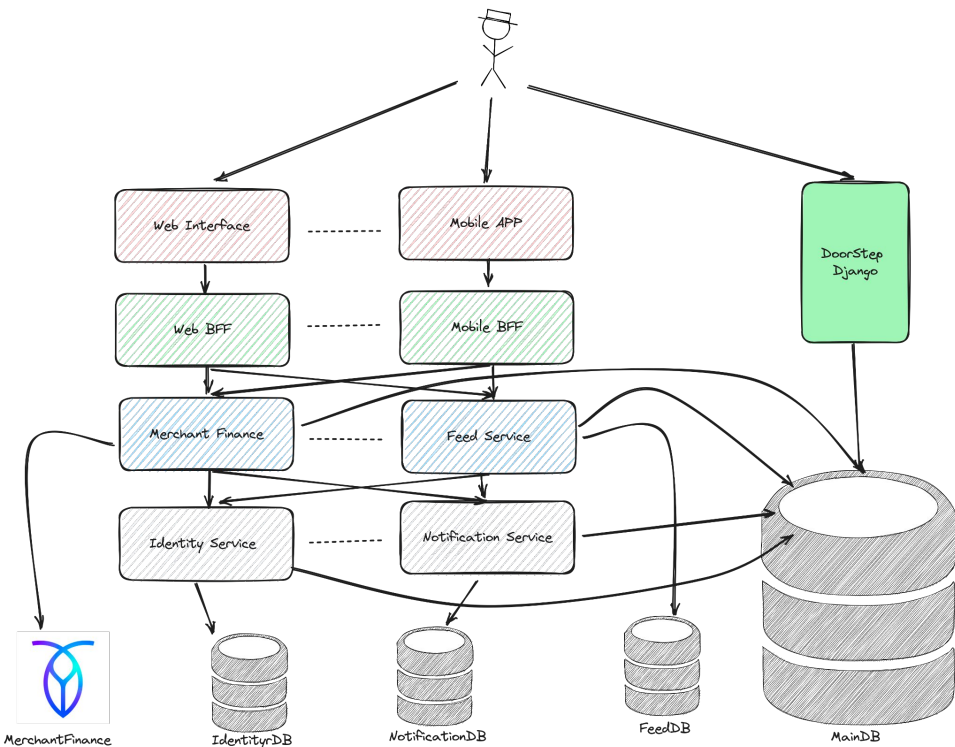
❖ Do not block microservices extractions

❖ Vertical federation

➤ extract tables to separate Aurora clusters

❖ Horizontal sharding

➤ migrate to CockroachDB





# WAKE-UP CALL: FRIDAY APRIL 17TH 2020



⇒ On 2020/04/17 MainDB peaked at 1.636 MQPS, leading to hours of downtime

## Ain't no rest for the wicked



**Franco Campilongo**  
Active 16m ago



**Franco Campilongo**  
You're friends on Facebook  
Owner-operator at iTalico and Owner-Operator at Terun  
Studied Economics at Università di Cosensa  
Lives in Redwood City, California

APR 18, 2020, 7:10 PM

ciao Franco

che issue state avendo con doordash?

Risolto ora  
Ci avevano messo on CALL orders only. Niente veniva sulla APP e siamo andati in 🍷



leri è stato un nightmare





APR 18, 2020, 7:10 PM

ciao Franco

che issue state avendo con doordash?

Risolto ora

Ci avevano messo on CALL orders only. Niente veniva sulla APP e siamo andati in 

Ieri è stato un nightmare



# WAKE-UP CALL: FRIDAY APRIL 17TH 2020



⇒ On 2020/04/17 MainDB peaked at 1.636 MQPS,  
leading to hours of downtime

## Ain't no rest for the wicked

⇒ Gained some quick headroom,  
then hacked together an extraction tool prototype, ready by end of April

⇒ Made 5 attempts at the first database extraction (Identity),  
4 reverted, 1 successful on 2020/05/22

⇒ In the 33 days since then, we extracted 7 databases & 54 tables!!!



# WAKE-UP CALL: FRIDAY APRIL 17TH 2020



⇒ On 2020/04/17 MainDB peaked at 1.636 MQPS, leading to hours of downtime

## Ain't no rest for the wicked

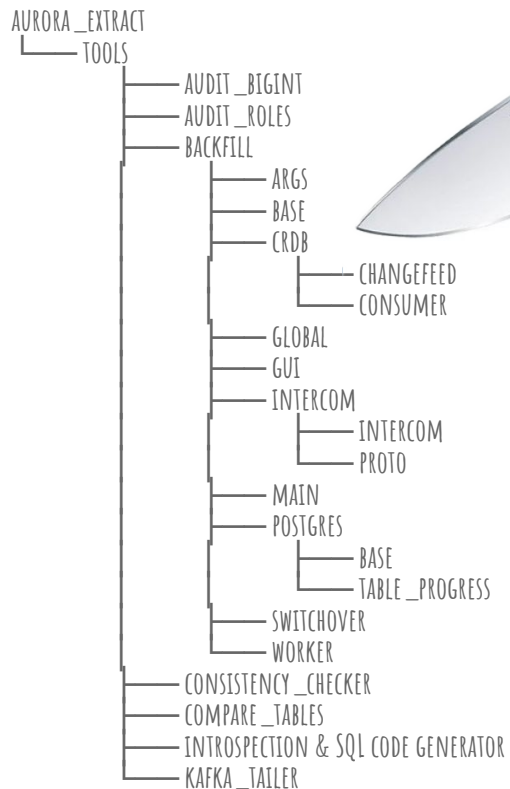
⇒ Gained some quick headroom, then hacked together an extraction tool prototype, ready by end of April

⇒ Made 5 attempts at the first database extraction (Identity), 4 reverted, 1 successful on 2020/05/22

⇒ In the 33 days since then, we extracted 7 databases & 54 tables!!!



# A COLLECTION OF TOOLS

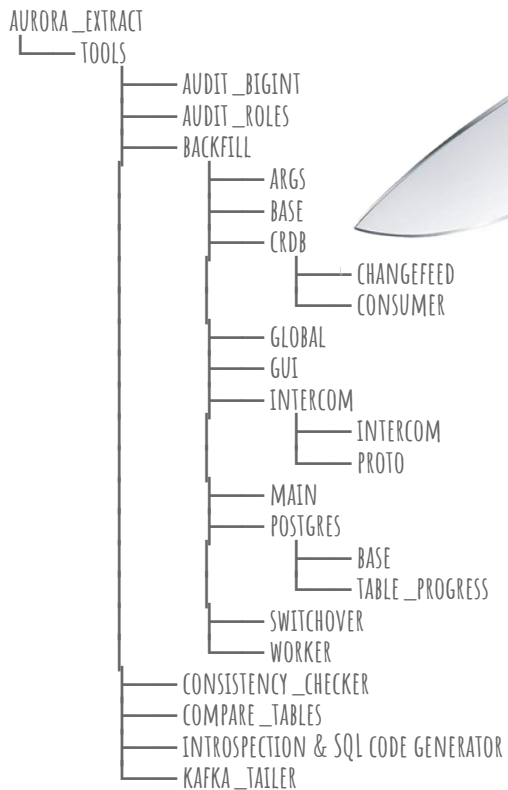


# MOTIVATION

TRADITIONAL TABLE EXTRACTION  
REQUIRED CHANGES IN EACH CLIENT  
SEVERAL ENGINEERS x 6 MONTHS  
BACKFILL + SYNCING  
VIA DOUBLE WRITES + DOUBLE READS  
INCREASED DATABASE LOAD  
NOT HALTABLE & RESUMABLE  
LABOR INTENSIVE  
ERROR PRONE

LED TO INCONSISTENCIES DIFFICULT TO RECONCILE  
TIGHTLY COUPLED SYNCHRONOUS PUSH MODEL

# A COLLECTION OF TOOLS



# MOTIVATION

BIGINT UPGRADES

REQUIRED TRIGGERS

LEAD TO SEVERAL LOCK PILEUPS

BACKFILL + SYNCING

VIA AFOREMENTIONED TRIGGERS

INCREASED DATABASE LOAD

NOT HALTABLE & RESUMABLE

ERROR PRONE

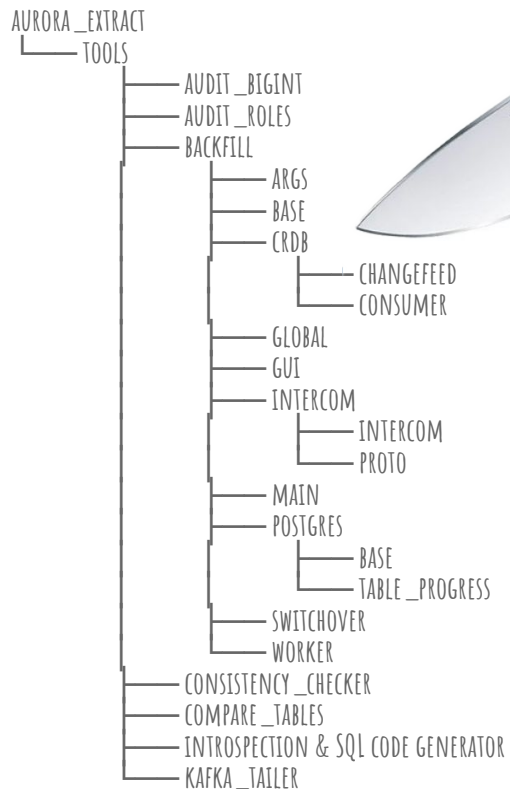
ATOMIC SWAP WAS OFTEN IMPEDED BY TRIGGERS

TIGHTLY COUPLED SYNCHRONOUS PUSH MODEL





# A COLLECTION OF TOOLS



# MOTIVATION

## RDS UPGRADES

FAR FROM HITLESS  
LEAD TO PROLONGED OUTAGES  
CLICK & PRAY

9.5 ⇒ 9.6 ⇒ 10 ⇒ 11 ⇒ 12  
ONE MAJOR VERSION AT A TIME

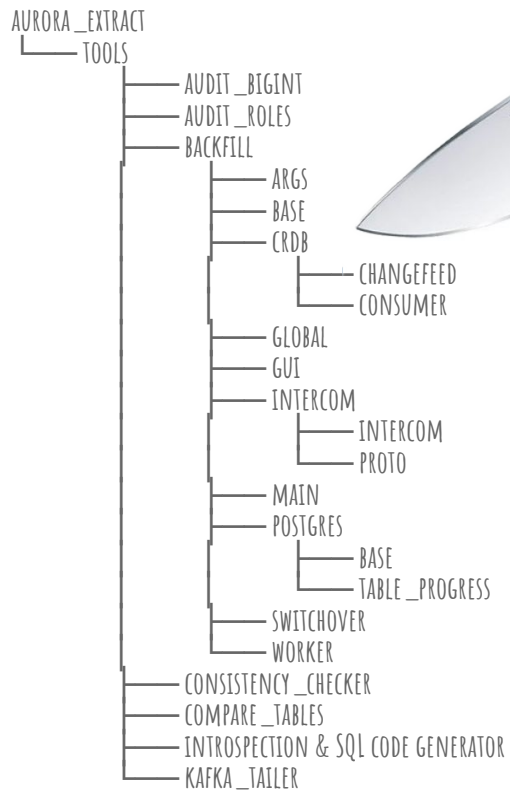
RDS ⇒ AURORA

ONE EXTRA STEP TO GO FROM PLAIN RDS TO RDS AURORA





# A COLLECTION OF TOOLS



# MOTIVATION

## CRDB EXTRACTIONS

HIGH ODDS A REVERT MIGHT BE REQUIRED  
NO (IN-HOUSE) «PRIOR ART»





# PHILOSOPHY: *Invest into reusable tools that would automate all of the most tricky bits*

- ⇒ *Minimize labor, never go through the same effort again, multiply the velocity at which we can proceed*
- ⇒ *Constantly improve and enhance the tool, invest on it and commoditize it via autotuning so any engineer can just fire it up*
- ⇒ *Provide a convenient UI that holds your hand during the operation and allows to edit the generated SQL*
- ⇒ *Add safeguards and autotuning, minimize the opportunities for human error*
- ⇒ *Allow for emergency revert in as few clicks as possible*
- ⇒ *Safely operate over production databases, both source and destination*



# TRADITIONAL MIGRATION PITFALLS

⇒ Aurora CDC: logical replication, 40% overhead on single primary  
not available on 9.6  
exposes to the danger of OOM killer

⇒ Triggers (for migrations) or Double Writes (for extractions)  
synchronous and tightly coupled,  
work can't be suspended and resumed as needed

⇒ Destination CPU overload & Aurora replica lag

Source contention and CPU overload

⇒ Heavy vacuuming or downtime due to transaction wraparound

⇒ Page buffer contention (Aurora) / hopping hotspots (CRDB)

⇒ High CRDB commit latency ⇒ large bandwidth-delay product

⇒ tail changes directly from tables

⇒ poll instead of push

⇒ feedback loop and PID controller

⇒ load balance writes across cluster

⇒ load balance reads across read replicas

`aurora_replica_status()` &  
`crdb_internal.kv_node_status` &  
`show ranges`

⇒ auto-tunable batch size

⇒ spread out batches/distant extents

⇒ fill BDP via massive parallelism



# KEEP IT SIMPLE STUPID!

⇒ *At any point in time, have a clear Source of Truth*

- *before the extraction, the SoT is the "old" database and the new database catches up*
- *after the extraction, the SoT is the "new" database and the old database catches up*

⇒ *There's no need or desire to replay each and every individual change*

- *we don't need to capture all changes, but just **what** changed (i.e. what was inserted, updated, or deleted)*
- *always fast-forward to the latest – will need a rolling max of the replication lag to have a truthful estimate*

⇒ *If possible, avoid pulling and pushing data via a client*

- *leverage foreign data wrappers*

⇒ *If/when we can't leverage foreign data wrappers,*

- *maximize parallelism so as to be able to fill the much greater bandwidth-delay product*

⇒ *Iterative looping & chunking*

⇒ *several python asyncio event loops with multi-processing*

*vs.*

⇒ *Declarative transformations, filtering & syncing*

⇒ *code-generated DML SQL* ❤️



# SOURCE DATABASE INTROSPECTION



# CODE

## GENERATION

### Postgres Source Setup DDL

- `add __updated_at` column if missing
- lightweight triggers
  - bypass idempotent writes
  - bump update timestamp only if client didn't
  - track deletes in outbox
- foreign servers (destination primary / replicas / CRDB)
- foreign tables
- delete outboxes
- unique constraint violation resolution logs/backups

### CRDB Destination Setup

- `table & sequence definitions`
- changefeeds
- unique constraint violation resolution logs/backups

- "Hybrid" PgBouncer setup derived automagically from regular source pgbouncer via code generation

### Postgres Destination Setup DDL

- `table & sequence definitions`
- lightweight triggers
  - bypass idempotent writes
  - bump update timestamp only if client didn't
  - track deletes in outbox
- foreign servers (source replicas)
- foreign tables
- delete outboxes
- unique constraint violation resolution logs/backups



# RUNTIME BEHAVIOR

Generate DML SQL that (optionally) accounts for:

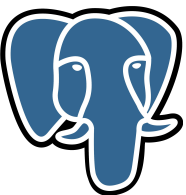
- ❖ *Changes in schema and required data transformations*
- ❖ *Changes in primary key*
- ❖ *Desired constraint to upsert by*
- ❖ *Any desired filtering*
  - ⚠ *careful about cardinality if not supported by appropriate index!*
- ❖ *Desired behavior*
  - *only clobber if more recent*
  - *only clobber if missing / just fill the gaps along the desired constraint*
  - *never clobber / just insert rows that do not violate any constraint*
    - ⇒ *useful to speed-up resolution*
  - *always clobber / repave*
  - *just compare the two databases for consistency*





store

|          |     |
|----------|-----|
| store_id | ... |
| store_id | ... |
| store_id | ... |
| store_id | ... |



# SCHEMA TRANSFORMATIONS

store

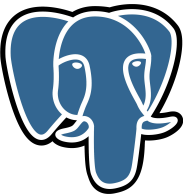
|             |          |     |
|-------------|----------|-----|
| merchant_id | store_id | ... |
|             | store_id | ... |
|             | store_id | ... |
|             | store_id | ... |
| merchant_id | store_id | ... |
|             | store_id | ... |





store

|          |     |             |
|----------|-----|-------------|
| store_id | ... | NULL        |
| store_id | ... | NULL        |
| store_id | ... | NULL        |
| store_id | ... | NULL        |
| store_id | ... | merchant_id |
| store_id | ... | merchant_id |



# SCHEMA TRANSFORMATIONS

store

|             |          |     |
|-------------|----------|-----|
| merchant_id | store_id | ... |
|             | store_id | ... |
|             | store_id | ... |
|             | store_id | ... |
| merchant_id | store_id | ... |
|             | store_id | ... |

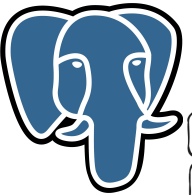




# SCHEMA TRANSFORMATIONS

store

|          |     |             |
|----------|-----|-------------|
| store_id | ... | NULL        |
| store_id | ... | NULL        |
| store_id | ... | NULL        |
| store_id | ... | NULL        |
| store_id | ... | merchant_id |
| store_id | ... | merchant_id |



lookup\_store2merchant

|          |             |
|----------|-------------|
| store_id | merchant_id |
| store_id | merchant_id |
| store_id | merchant_id |
| store_id | merchant_id |

store

|             |          |     |
|-------------|----------|-----|
| merchant_id | store_id | ... |
|             | store_id | ... |
|             | store_id | ... |
|             | store_id | ... |
| merchant_id | store_id | ... |
|             | store_id | ... |

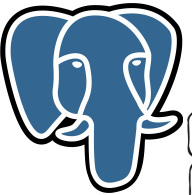




# SCHEMA TRANSFORMATIONS

store

|          |     |             |
|----------|-----|-------------|
| store_id | ... | NULL        |
| store_id | ... | NULL        |
| store_id | ... | NULL        |
| store_id | ... | NULL        |
| store_id | ... | merchant_id |
| store_id | ... | merchant_id |



lookup\_store2merchant

|          |             |
|----------|-------------|
| store_id | merchant_id |
| store_id | merchant_id |
| store_id | merchant_id |
| store_id | merchant_id |

```
SELECT
  COALESCE(
    s.merchant_id,
    s2m.merchant_id,
    0),
  ...
FROM store s
LEFT JOIN lookup_store2merchant s2m
ON s.store_id = s2m.store_id
WHERE ...;
```

```
INSERT INTO store(
  merchant_id,
  store_id,
  ...)
VALUES ...;
```



store

|             |          |     |
|-------------|----------|-----|
| merchant_id | store_id | ... |
|             | store_id | ... |
|             | store_id | ... |
|             | store_id | ... |
| merchant_id | store_id | ... |
|             | store_id | ... |



# RUNTIME BEHAVIOR

2/2

Iteratively issue DML SQL against far-apart chunks of:

- ❖ range key / circular key (i.e. random integer or UUID) / timestamp (i.e. created\_at or updated\_at).
  - all mapped to integers for simplicity to keep a single, coherent implementation
- ❖ deletes captured in delete outbox
- ❖ primary keys from changefeed

If unable to SET SESSION\_REPLICATION\_ROLE = 'REPLICA', catch & resolve constraint violations automatically as you go.

Perform careful checkpointing, so as to be able to pause and resume the initial import.

Keep catching up after initial import.

 careful with high watermarks: account for the latency between a write and its commit!

Finally, cut the source of truth over!

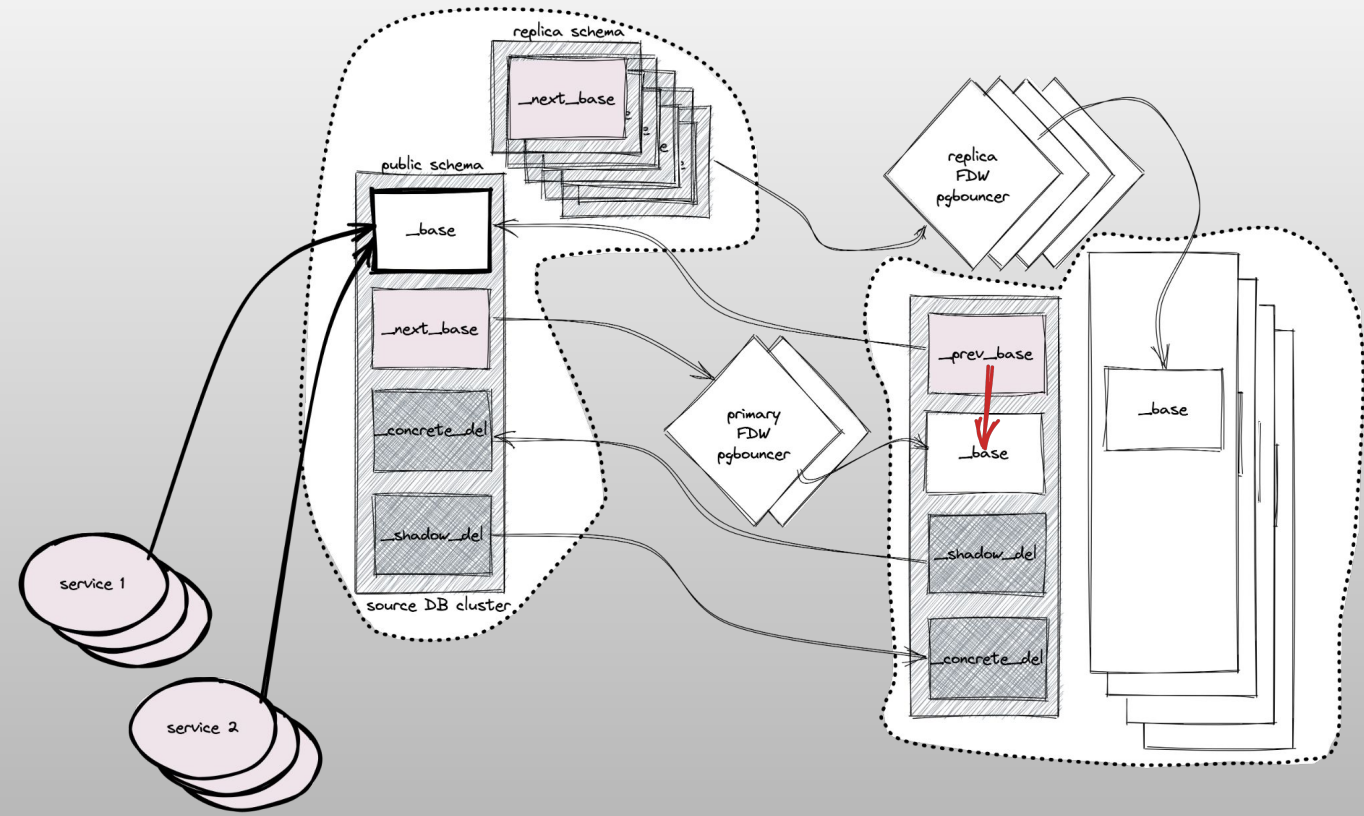


CAROSSELLO



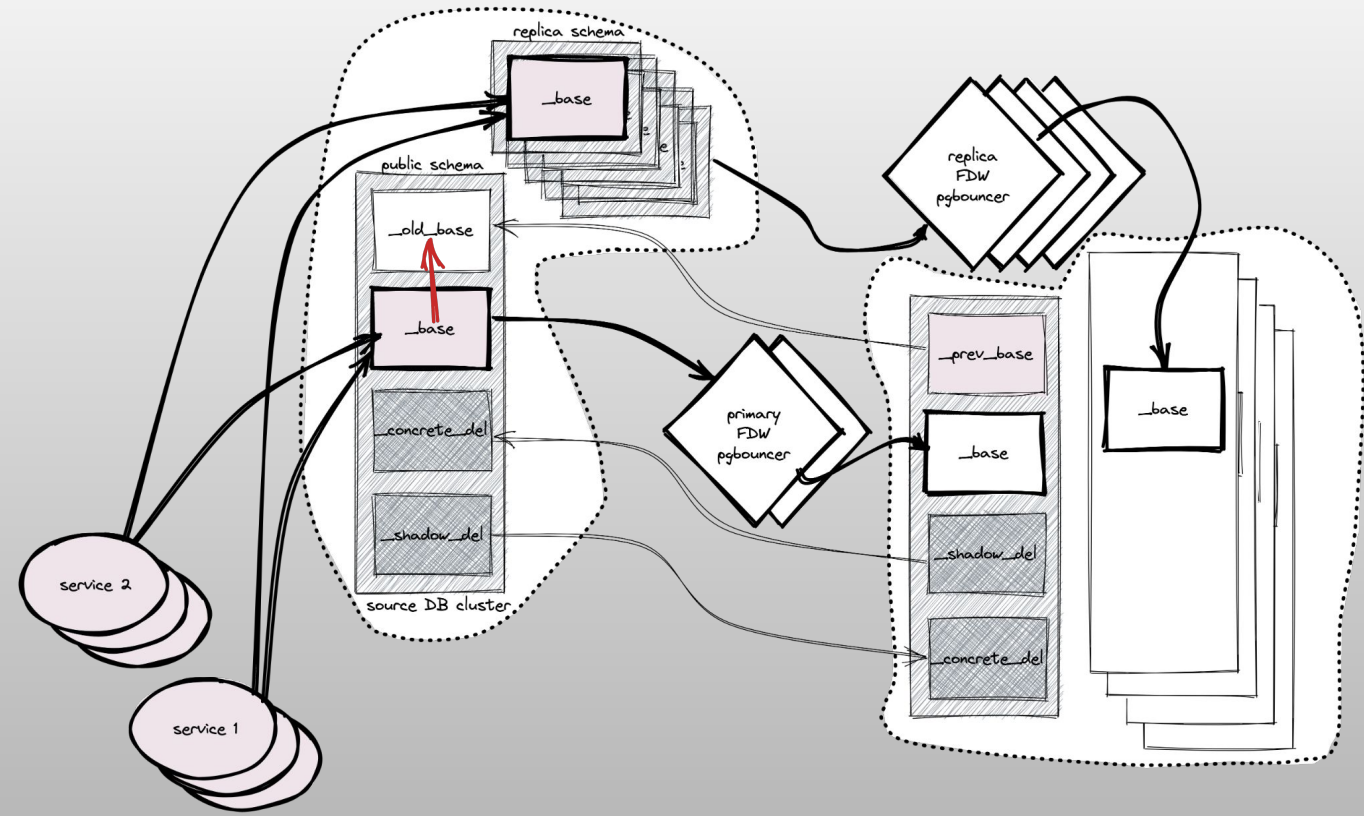


# TABLE EXTRACTION TO (AURORA) POSTGRES: *before cutover*



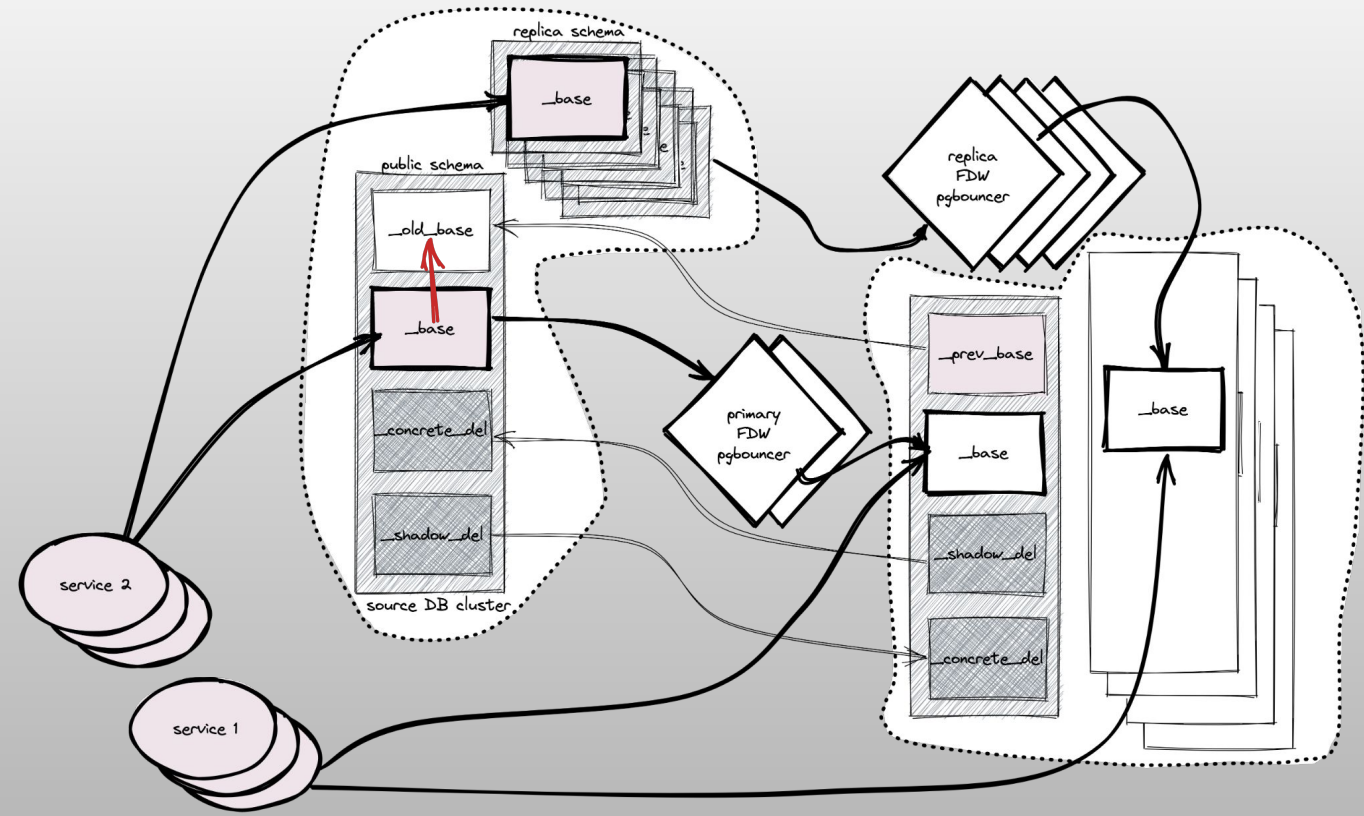


# TABLE EXTRACTION TO (AURORA) POSTGRES: *after cutover*



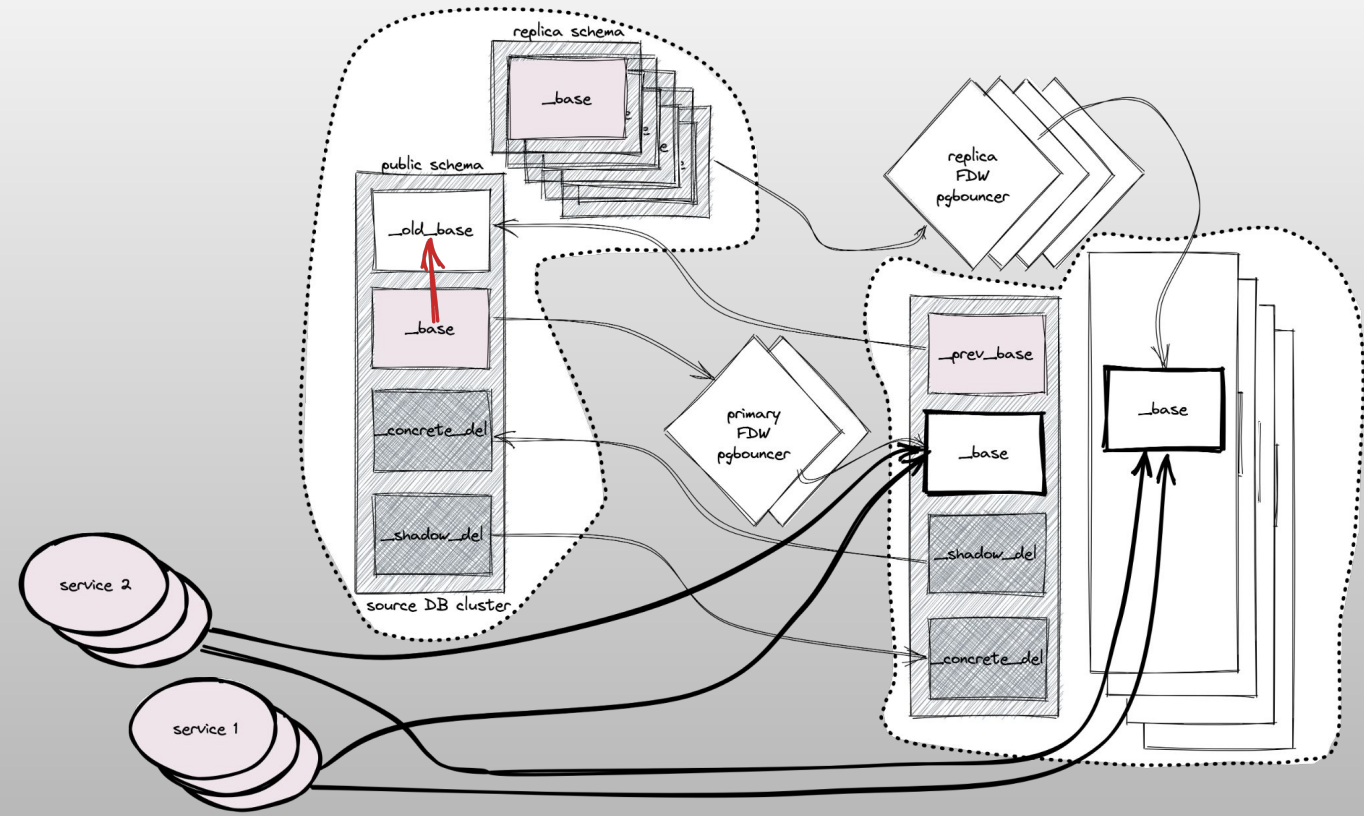


# TABLE EXTRACTION TO (AURORA) POSTGRES: *after cutover*



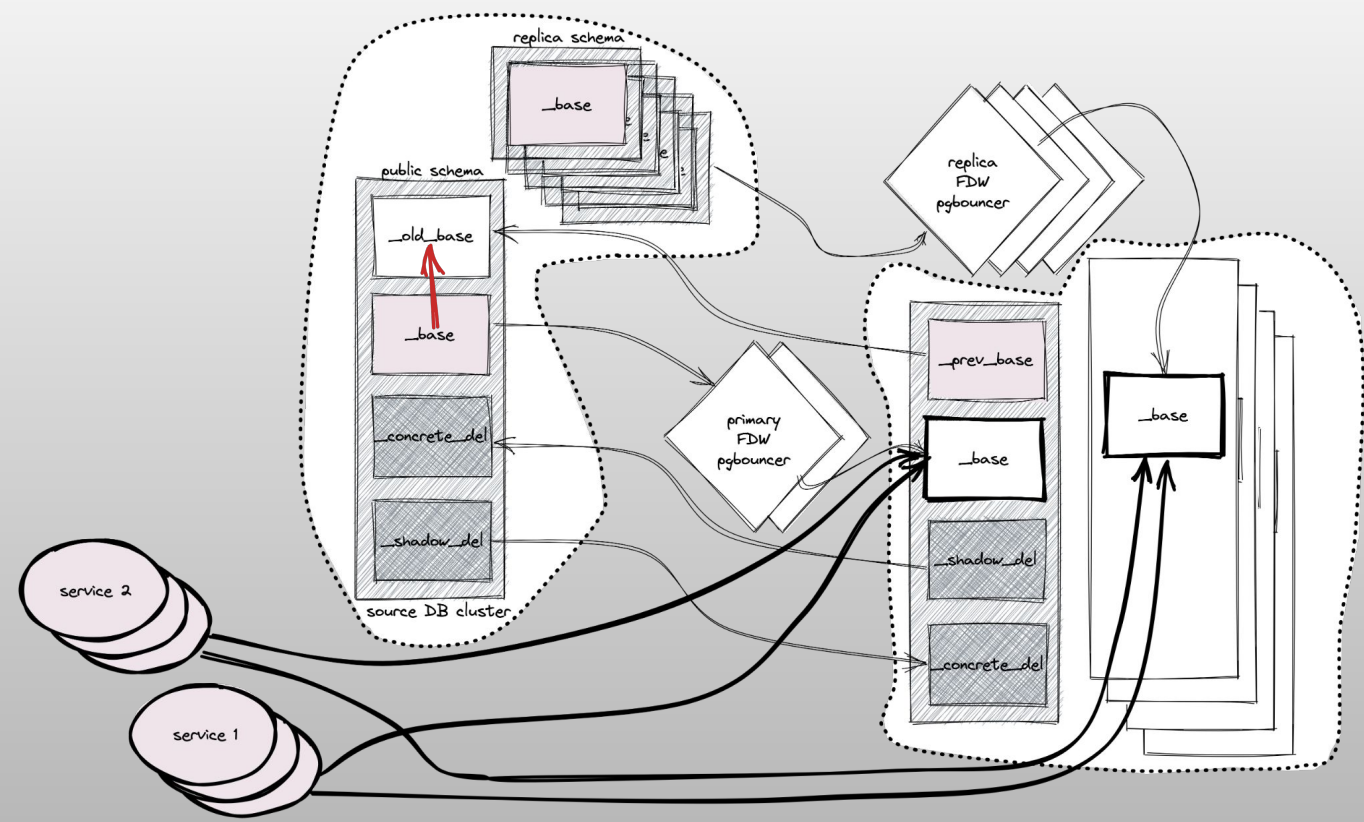


# TABLE EXTRACTION TO (AURORA) POSTGRES: *after cutover*





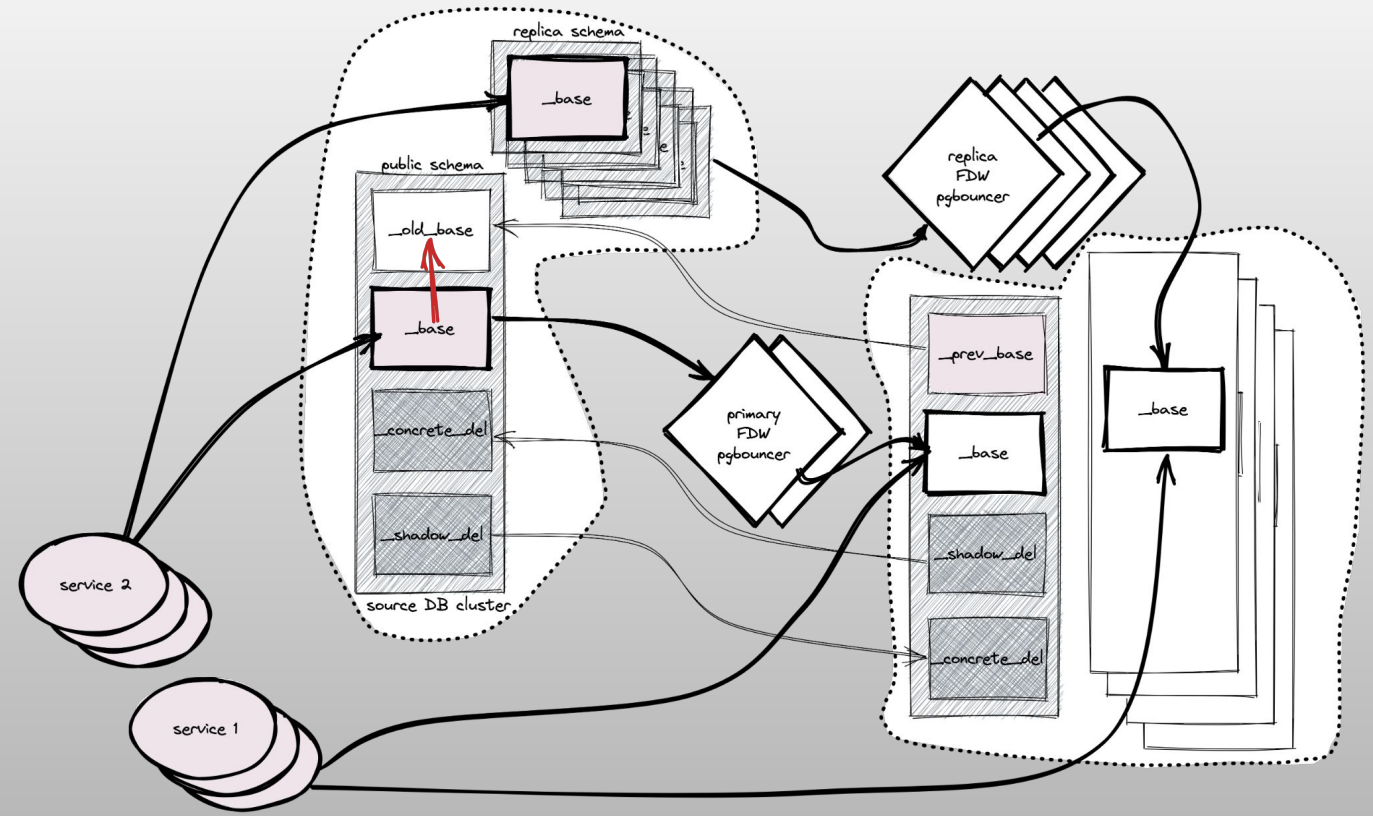
# TABLE EXTRACTION TO (AURORA) POSTGRES: *before revert*





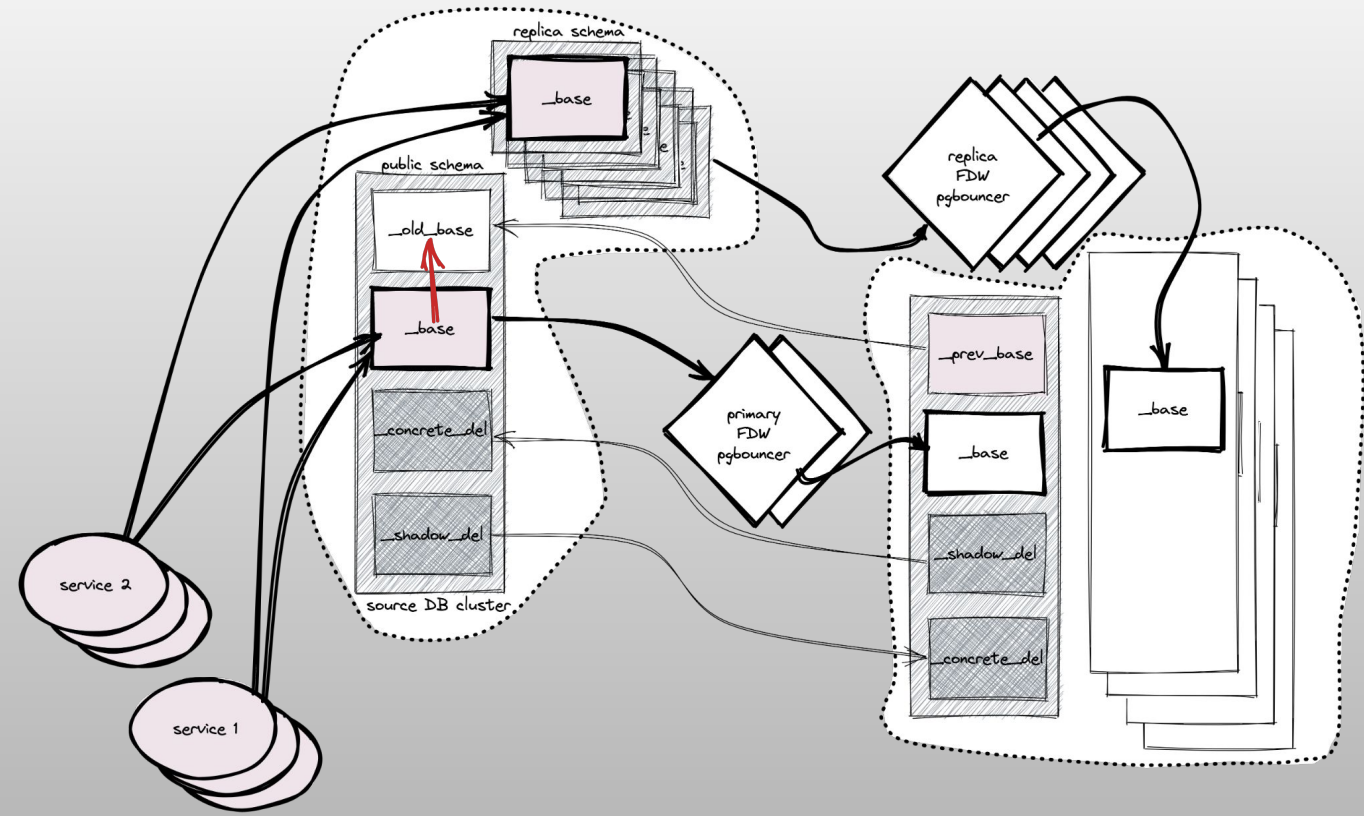


# TABLE EXTRACTION TO (AURORA) POSTGRES: *before revert*





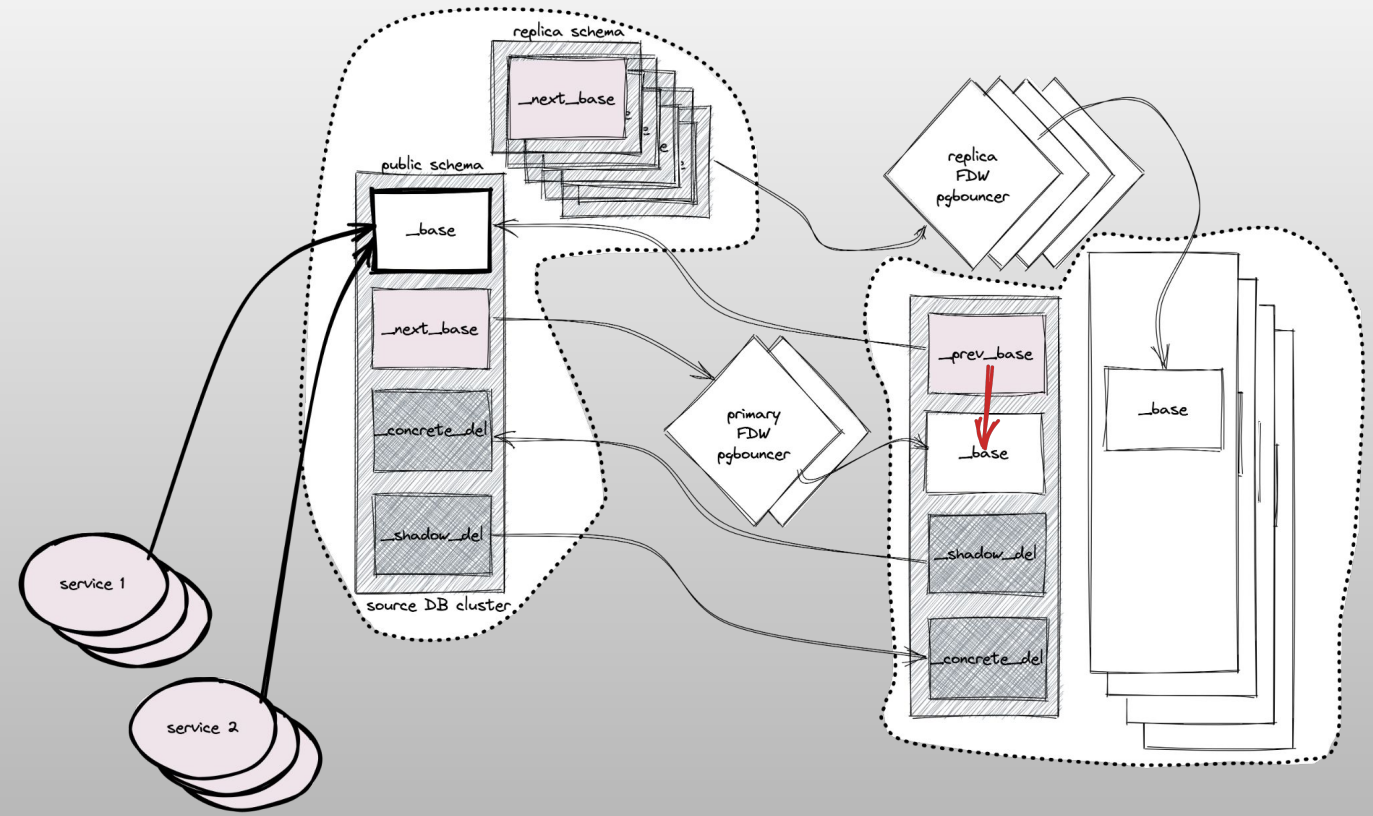
# TABLE EXTRACTION TO (AURORA) POSTGRES: *before revert*







# TABLE EXTRACTION TO (AURORA) POSTGRES: *after revert*

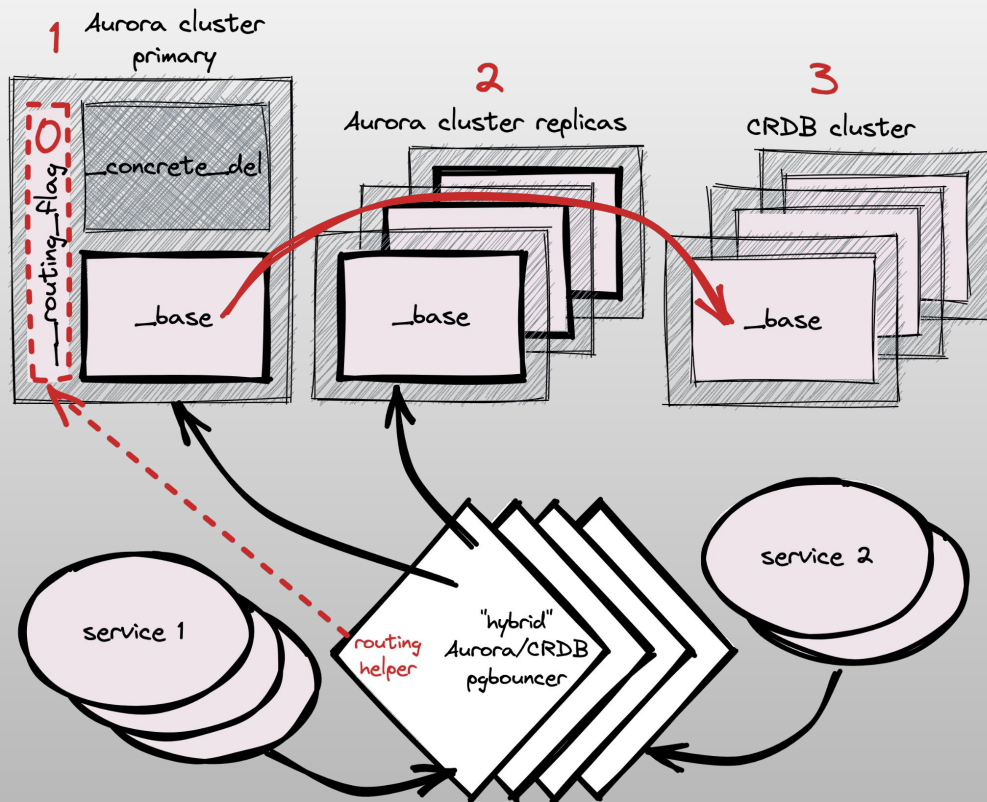




*Intervallo*

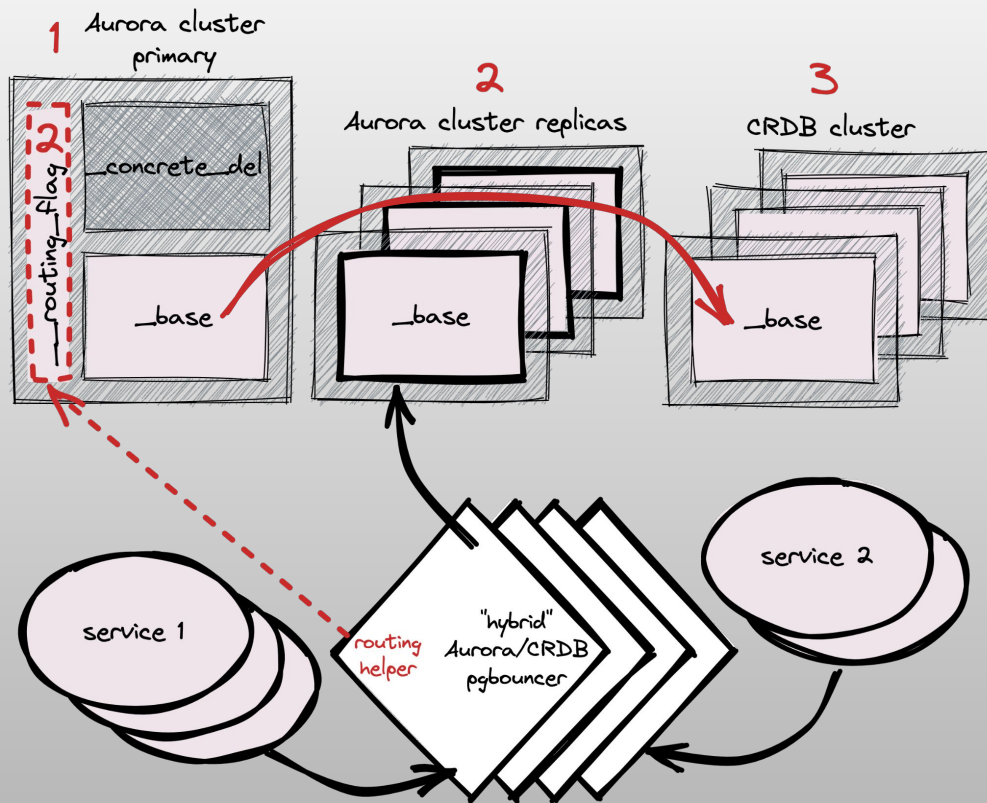
# FORKLIFT EXTRACTION TO CRDB:

before cutover



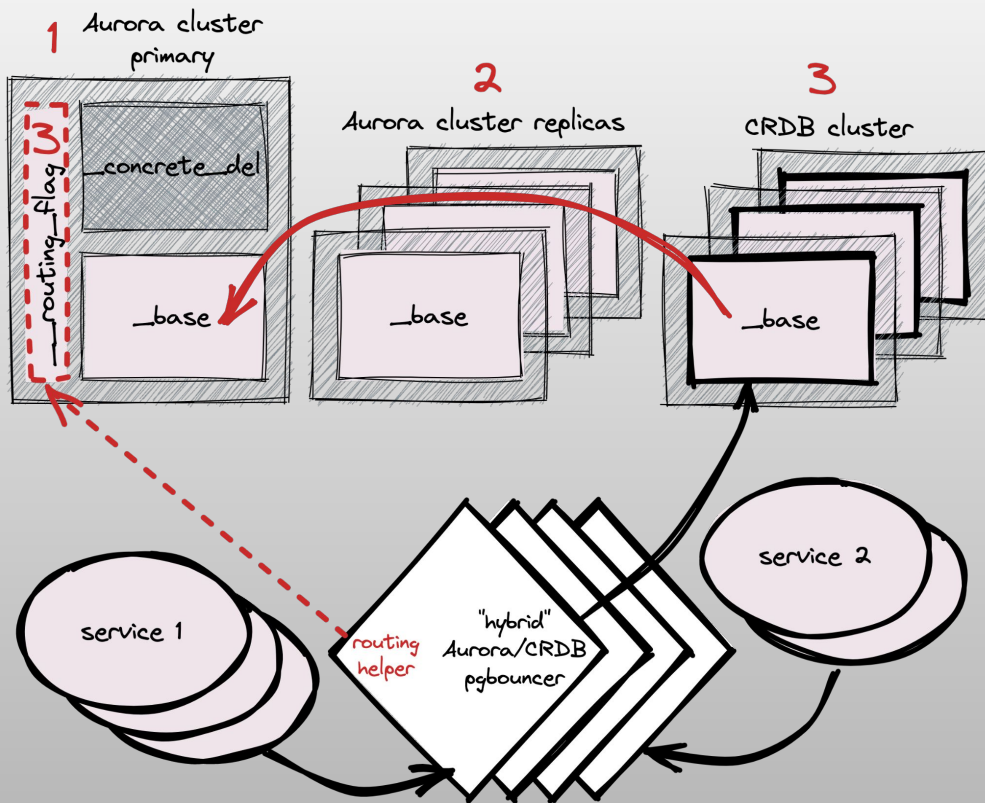
# FORKLIFT EXTRACTION TO CRDB:

*cutover*



# FORKLIFT EXTRACTION TO CRDB:

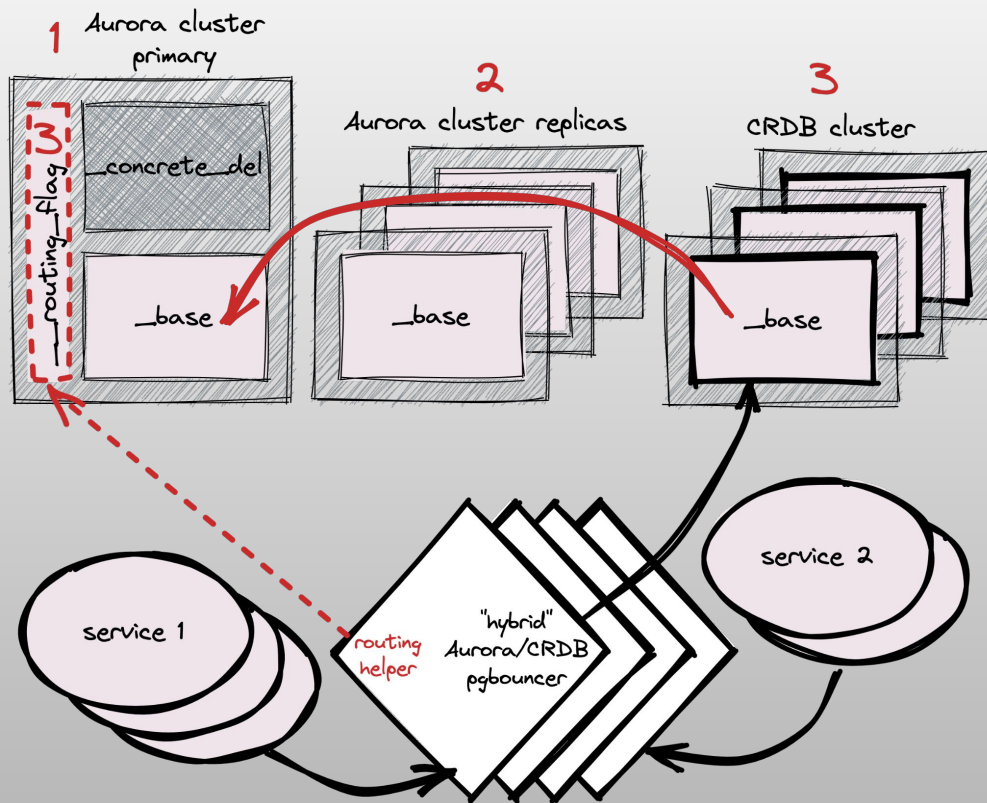
after cutover





# FORKLIFT EXTRACTION TO CRDB:

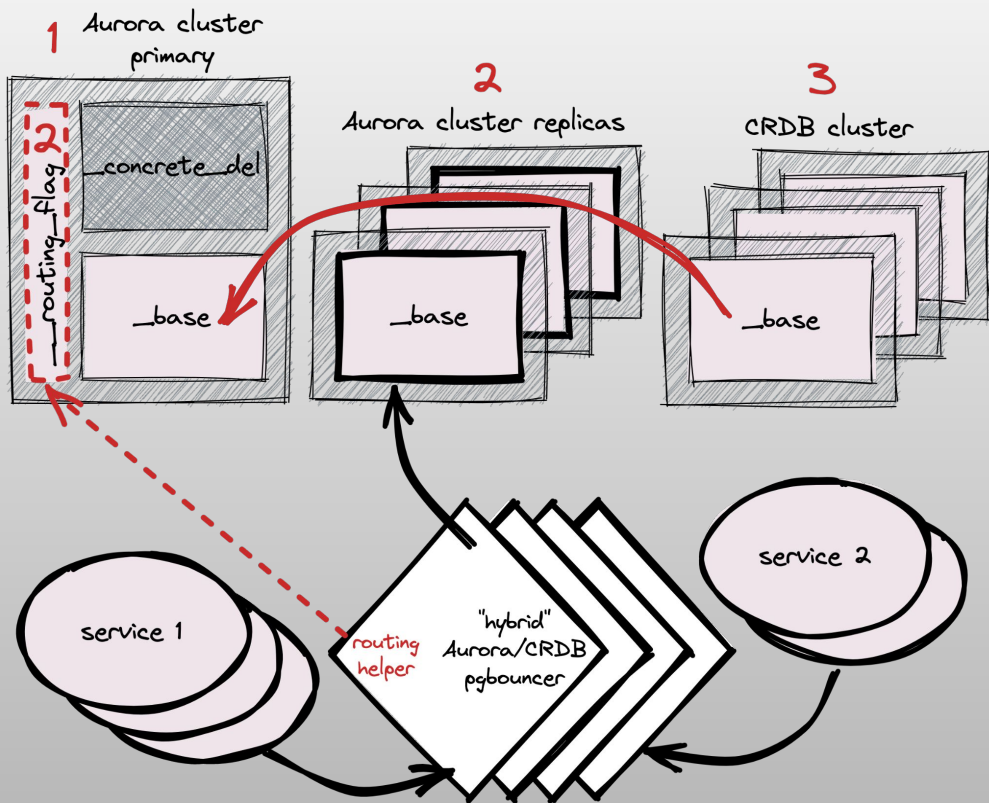
*before revert*





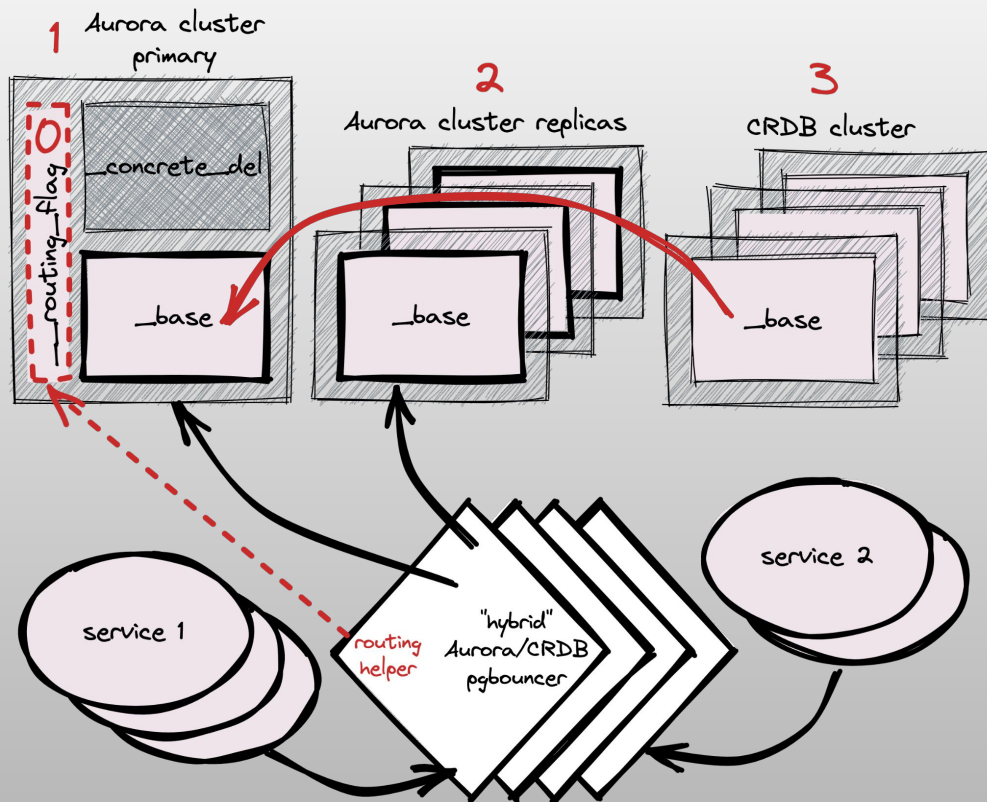
# FORKLIFT EXTRACTION TO CRDB:

revert



# FORKLIFT EXTRACTION TO CRDB:




after revert





aurora-extract APP 06:50

Thursday, March 17th ▾

Extracting tables   

Extracting `delivery` from `OldDeliveryDB` to `DeliveryCRDB`

speak now on [921-6392-3802](tel:921-6392-3802) or forever hold your peace

04:50:25 source of truth is `Aurora` armed and ready

04:52:51 source of truth is `Aurora` checkpointed kafka topics to minimize revert time

04:52:55 source of truth is `Aurora` writes disabled

04:53:20 source of truth is `CRDB` writes enabled

04:53:25 source of truth is `CRDB` disarmed

04:54:50 source of truth is `CRDB` armed and ready

04:55:21 source of truth is `CRDB` writes disabled

04:55:32 source of truth is `Aurora` writes enabled

04:55:54 source of truth is `Aurora` disarmed

25s

11s



aurora-extract APP 08:14

Tuesday, March 29th ▾

Extracting tables   

Extracting `delivery` from `OldDeliveryDB` to `DeliveryCRDB`

speak now on [921-6392-3802](tel:921-6392-3802) or forever hold your peace

05:14:46 source of truth is `Aurora` armed and ready

05:16:10 source of truth is `Aurora` checkpointed kafka topics to minimize revert time

05:16:15 source of truth is `Aurora` writes disabled

05:16:35 source of truth is `CRDB` writes enabled

05:16:58 source of truth is `CRDB` disarmed

05:29:12 source of truth is `CRDB` armed and ready

05:29:22 source of truth is `CRDB` writes disabled

05:29:33 source of truth is `Aurora` writes enabled

05:29:46 source of truth is `Aurora` disarmed

20s

11s

Thread

# eng-oncall ☰



aurora-extract APP Mar 30th at 07:30

Extracting tables   

Extracting `delivery` from `OldDeliveryDB` to `DeliveryCRDB`

speak now on [921-6392-3802](tel:921-6392-3802) or forever hold your peace

04:30:22 source of truth is `Aurora` armed and ready

04:30:30 source of truth is `Aurora` checkpointed kafka topics to minimize revert time

04:30:34 source of truth is `Aurora` writes disabled

04:31:02 source of truth is `CRDB` writes enabled

04:31:18 source of truth is `CRDB` disarmed

28s

 1 

8 replies



Sebastian Yates  3 months ago



Karthik Katooru 3 months ago

(celebration after a peak or two)

 1 



alessandro.salvatori 3 months ago

looks like it



alessandro.salvatori 3 months ago



Sebastian Yates  3 months ago

Exciting



Rohan Chopra 3 months ago



Aditya Bipinchandra Parikh 3 months ago



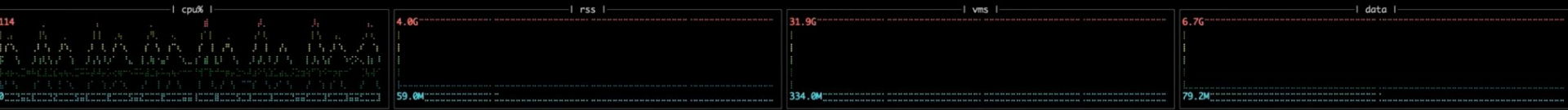


Ledger (LOG/gui-backfill-2022-02-15-06-34-29.log)

```

1 SWITCHOVER_STREAMING_SERVICE ERROR [-] 96885 [-] (NOT AN ERROR) polling routing flag from "__routing_flag_delivery_order_info"
2
3

```



client-side contention avoidance  
 client-side deadlock avoidance

Options |

<FORCE ARM >      (A) Aurora with writes enabled  
 (D) Aurora with writes disabled  
 (C) CRDB

Tiers |

] +/- 1x Sleep q Abort meTrics m/M x/2 g/G 2.5 u/U 1x x Send Control-C. l layout ::: replay\_latency    0 r/Replica lag 9/20/60/85 c/Cpu 10%/45 ::: delivery\_order\_info 2

THIS ITALIAN NEO-REALIST FILM IS REALLY LONG.  
CAN IT BE DONE ALREADY?

*Fine*

