PCR Replication

Physical Cluster Replication (PCR) asynchronously replicates data/metadata from a primary cluster (active, serving apps) to a standby (passive, for DR/failover).

Setup Instruction:

On Primary (Existing Cluster)

Enable rangefeed:

```
SET CLUSTER SETTING kv.rangefeed.enabled = true;
```

Create replication user:

```
CREATE USER <repl-user> WITH PASSWORD '<secure-password>';
GRANT SYSTEM REPLICATION TO <repl-user>;
```

Deploy New Standby Cluster

Generate certs for standby (match primary's CA)

Start nodes with virtualization

On Stand-by Node 1

```
cockroach start \
  --certs-dir=certs \
  --host=<standby-node1> \
  --ports=26257:8080 \
  --join=<standby-node1>,<standby-node2>,<standby-node3>:26257 \
  --virtualized-empty
```

Repeat for Nodes 2 & 3

Initialize and connect to new cluster

```
cockroach init --certs-dir=certs --host=<standby-node1> --virtualized-empty
cockroach sql    --url "postgresql://root@<standby-node1>:26257?options=-ccluster=system&sslmode=verify-full"   --certs-dir=certs
```

Enable rangefeed on standby

SET CLUSTER SETTING kv.rangefeed.enabled = true;

On Stand by Create a replication user:

```
CREATE USER <repl-user> WITH PASSWORD '<secure-password>';  -- Same as primary
GRANT SYSTEM REPLICATION TO <repl-user>;


SHOW VIRTUAL CLUSTERS;
```

On standby host, generate URI to primary (embeds primary CA):

```
cockroach encode-uri   <repl-user> :<secure-password> @<primary-node-ip>:26257   --cacert=<path-to-primary-certs>/ca.crt  --inline
```

Start Replication Stream

```
CREATE VIRTUAL CLUSTER main FROM REPLICATION OF system ON '<full-uri-from-previous-step>';


SHOW VIRTUAL CLUSTERS;
```

Monitor:

- Jobs: `SHOW JOBS;` (look for replication consumer/producer).
- Lag: `SELECT * FROM crdb_internal.physical_replication_streams;`
- UI: DB Console > Virtual Clusters (on standby).

For failover:

On standby:

ALTER VIRTUAL CLUSTER main FAILOVER TO STANDBY; then ALTER VIRTUAL CLUSTER main START SERVICE;