Multi Region Clusters

Locality settings in CockroachDB enable multi-region deployments by assigning geographic constraints (regions, availability zones) to nodes, databases, and tables. This optimizes data placement, latency, and compliance (e.g., data domiciling). Requires a multi-region cluster with node localities defined at startup.

Nodes

- **Purpose**: Defines the node's geographic location for data placement and query routing.
- **Configuration**:
  - Set at startup: cockroach start --locality=region=us-west,zone=west1 ...
  - Multiple nodes: Assign unique combinations (e.g., 3 AZs per region for HA).
  - View: `SHOW LOCALITY;` (returns empty if unset).
- **Best Practices**: Use for fault tolerance (e.g., 3 regions/AZs); enable Node Map with lat/long for topology views.
- **Impact**: Influences range allocation; rebalancing prefers local replicas.

```
cockroach start \
  --certs-dir=${COCKROACHDB_CERTS_DIR} \
  --locality=region=us-east1,zone=a \
  --join=existing-node:26257,other-node:26257
```

Databases

- **Purpose**: Groups regions for multi-region ops; controls primary region for defaults and survival goals.
- **Configuration**:
  - Create: CREATE DATABASE multiregion PRIMARY REGION "us-east1" REGIONS "us-east1", "us-west1", "eu-west1";
  - Add region: ALTER DATABASE multiregion ADD REGION "eu-west1";
  - Set primary: ALTER DATABASE multiregion SET PRIMARY REGION "eu-west1";
  - Super regions (for domiciling): ALTER DATABASE multiregion ADD SUPER REGION eu_super WITH REGIONS ("eu-west1", "eu-central1");—restricts regional tables to super region nodes.
  - Restricted placement: ALTER DATABASE multiregion SET PLACEMENT RESTRICTED; (blocks non-local inserts).
- **Defaults**: Tables inherit PRIMARY REGION unless specified.

Tables

- **Purpose**: Optimizes data access (reads/writes) by homing data to specific regions; types: Regional (low-latency in one region), Global (low-latency everywhere).

Types and Configuration

Type	Syntax	Description	Use Case
REGIONAL BY TABLE	LOCALITY REGIONAL BY TABLE IN "us-east1"	Homes entire table to one region (default: PRIMARY REGION).	Regional apps; low latency in home region.
GLOBAL	LOCALITY GLOBAL	Replicates data everywhere; optimized for multi-region reads.	Global lookups (e.g., users table).

Examples:

ALTER TABLE orders SET LOCALITY GLOBAL;

ALTER TABLE users SET LOCALITY REGIONAL BY TABLE IN "us-east1";

ALTER TABLE users SET LOCALITY REGIONAL BY TABLE IN PRIMARY REGION;