CockroachDB中，用户可以使用复制区域（**Replication Zones**）来控制特定数据的冗余副本的数量和位置，包括在首次添加的副本，或是重新平衡以维持群集中均衡的副本。 默认情况下，内部系统数据有一些特殊的预配置的复制区域，而群集其余部分数据在的一个默认复制区域。 用户可以根据需要调整这些预配置区域以及为单个数据库、表、行（[企业版特性](http://doc.cockroachchina.baidu.com/#deploy/enterprise-licensing/)）添加区域。例如，用户可以使用默认区域冗余集群中大多数数据在该复制区域中，同时创建特定复制区域以跨多个数据中心和地理位置的形式复制某个特定的数据库或表。

本小节将展示复制区域的工作原理以及如何使用cockroach zone命令配置它们。

**NOTE:** 目前只有root用户才能配置复制区域。

概述

复制区域级别

集群中表数据有4个复制区域级别：

| **级别** | **简介** |
| --- | --- |
| Cluster | CockroachDB预配置了复制区域.default，该区域适用于集群中的所有表数据，不受数据库、表或行特定复制区域的约束。此区域可以调整但不能删除。详见[查看默认复制区域](http://doc.cockroachchina.baidu.com/#deploy/configure-replication-zones/#view-the-default-replication-zone)和[编辑默认复制区域](http://doc.cockroachchina.baidu.com/#deploy/configure-replication-zones/#edit-the-default-replication-zone)。 |
| Database | 用户可以针对某个特定的数据库添加复制区域，详见[为数据库创建复制区域](http://doc.cockroachchina.baidu.com/#deploy/configure-replication-zones/#create-a-replication-zone-for-a-database) |
| Table | 用户可以针对某个特定的表添加复制区域，详见[为表创建复制区域](http://doc.cockroachchina.baidu.com/#deploy/configure-replication-zones/#create-a-replication-zone-for-a-table)  CockroachDB为内部表system.jobs预配置了一个复制区域，该表记录长时间运行的作业（如Schema更改和备份）的元数据。CockroachDB不会对此表进行历史查询，其中的行会频繁更新，因此预配置区域会为此表提供低于默认值的ttlseconds。 |
| Row（企业版特性） | 用户可以通过[定义分区](http://doc.cockroachchina.baidu.com/#deploy/define-table-partitions-(enterprise)/)，针对某些特定的行添加复制区域，详见[为行创建复制区域](http://doc.cockroachchina.baidu.com/#deploy/configure-replication-zones/#create-a-replication-zone-for-a-table-partition-new-in-v2-0) |

此外，CockroachDB将内部系统数据存储在系统range内。 内部系统数据对应两个复制区域级别：

| **等级** | **简介** |
| --- | --- |
| Cluster | 复制区域.default也适用于不受更具体的复制区域约束的所有系统Ranges。 |
| System Range | CockroachDB为“meta”和“liveness”系统Ranges提供了预配置的复制区域。如有必要，用户还可以为“timeseries”range和其他“系统”range添加复制区域。详见[为系统范围添加复制区域](http://doc.cockroachchina.baidu.com/#deploy/configure-replication-zones/#create-a-replication-zone-for-a-system-range)。 |

在复制表数据或系统数据时，CockroachDB始终使用最小粒度、可用的复制区域。例如，对于一条用户数据：

* 如果该行有一个复制区域，CockroachDB会使用它。
* 如果没有适用的行复制区域，CockroachDB将使用表复制区域。
* 如果没有适用的表复制区域，CockroachDB将使用数据库复制区域。
* 如果没有适用的数据库复制区域，CockroachDB将使用.default集群范围的复制区域。

**WARNING:** 修改.default复制区域不会应用到其他已有的复制区域。如果用户需要增加.default的冗余副本数，用户还需要增加[必要内部数据](http://doc.cockroachchina.baidu.com/#deploy/configure-replication-zones/#create-a-replication-zone-for-a-system-range)的冗余副本数。

复制区域格式

YAML配置文件中复制区域的相关配置如下：

range\_min\_bytes: <size-in-**bytes**>

range\_max\_bytes: <size-in-**bytes**>

gc:

ttlseconds: <time-in-seconds>

num\_replicas: <number-of-replicas>

constraints: <json-formatted-constraints>

| **配置项** | **简介** |
| --- | --- |
| range\_min\_bytes | 未实现 |
| range\_max\_bytes | 最大的range大小，以byte为单位。当一个Range的大小超过阈值的时候，Range会进行分裂操作。 **默认**：67108864（64MiB） |
| ttlseconds | 在进行垃圾回收之前，保留更新的数据的秒数。如果数据经常更新，则较小的ttlseconds值可以节省磁盘空间; 而较大的值会增加AS OF SYSTEM TIME所查询的范围，也称为[Time Travel Queries](http://doc.cockroachchina.baidu.com/#develop/sql-statements/SELECT/#select-historical-data-time-travel)。  建议不要将其设置为600（对应10分钟）以下，否则会导致长时间运行的查询出现问题。此外考虑懂啊一个记录所有版本数据应该存储在同一个Range内，因此不建议将其设置得过高，以至于某一时间段内对一个记录的所有更新的数据大小总计超过64MiB，此时可能导致服务器内存不足或诱发其他问题。 **默认**：90000（25小时） |
| num\_replicas | 一个zone内数据冗余的副本数量。 **默认**：3 |
| constraints | 一个JSON对象或数组用于约束副本的位置。详见[约束类型](http://doc.cockroachchina.baidu.com/#deploy/configure-replication-zones/#types-of-constraints)和[约束范围](http://doc.cockroachchina.baidu.com/#deploy/configure-replication-zones/#scope-of-constraints)。 **默认**：无约束，即CockroachDB将每个副本放在一个唯一的节点上，节点可能跨区域。 |

复制约束

副本的位置是基于分配给节点的描述性属性与区域配置中设置的约束。

**TIPS:** 有关如何在不同方案中设置节点属性和复制约束的演示，请参阅下面的基于场景的示例。

分配给节点的描述性属性

在执行cockroach start命令启动节点时，用户可以指定以下描述性属性：

| **属性类型** | **简介** |
| --- | --- |
| **Node Locality** | 使用--localityFlag，将一个描述节点位置的任意键值对分配给节点。位置可能包括国家、地区、数据中心、机架等。键值对内容的层次，其前后顺序应从大范围到小范围排列（例如先指定国家/地区，再指定数据中心，最后指定机架），所有节点键值和键值对的顺序必须是一样的。通常情况下键值对层次越多越好。例如： --locality=region=east,datacenter=us-east-1 --locality=region=east,datacenter=us-east-2 --locality=region=west,datacenter=us-west-1  CockroachDB按键值对内容的层次顺序，尝试根据位置在集群中均匀地散布副本。用户也可以配置复制区域来配合--locality使用，能够约束数据副本散布在特定的位置区域。  当节点之间存在较高延迟的时候，CockroachDB还使用--locality内容来优化租约持有者移动的范围，使其更接近当前客户端工作负载的热点位置，从而减少网络往返次数，提高读取性能。 |
| **Node Capability** | 使用--attrsFlag指定节点机器性能情况，包括特定硬件条件或CPU核心数量等。例如：--attrs=ram:64gb |
| **Store Type/Capability** | 使用--storeFlag中的attrs配置项指定节点硬盘类型或容量。例如： --store=path=/mnt/ssd01,attrs=ssd --store=path=/mnt/hda1,attrs=hdd:7200rpm |

约束类型

上述节点级和存储级描述性属性可以用于以下类型的复制区域约束，以控制副本的位置。同时注意：

* 若用户只是根据位置决定副本的位置，推荐不使用复制区域的约束。在没有区域约束的情况下，CockroachDB会尝试根据位置在集群中均匀地散布副本。
* 约束在一些特殊情况下非常有用，例如约束数据必须或不得存储在特定国家或特定类型的机器上。

| **约束类型** | **简介** | **句法** |
| --- | --- | --- |
| **Required** | 集群在散布Range副本的时候会将数据分发到特定位置或拥有特定属性的节点或存储设备上，不会存储数据在其他不符合条件的地方。 | +ssd |
| **Prohibited** | 集群在散布Range副本的时候会 **避免** 将数据分发到特定位置或拥有特定属性的节点或存储设备上，选择其他的位置。 | -ssd |

约束范围

用户可以指定约束，使它们适用于区域中的所有副本，或者指定不同的约束适用于不同的副本，即选择每个副本的确切位置。

| **约束范围** | **简介** | **句法** |
| --- | --- | --- |
| All Replica | 应用约束于复制区域的每个Range的所有副本 | constraints: [+ssd, -region=west] |
| Per-Replica | 在JSON对象中提供多个约束列表，约束列表将指定每个Range在不同区域的副本数量。Range副本总数不能大于配置的每个区域副本数的和值。 | constraints: {"+ssd,-region=west": 2, "+region=east": 1} |

节点/副本建议

详见生产部署当中[集群拓扑结构](http://doc.cockroachchina.baidu.com/#deploy/production-checklist/#cluster-topology)的相关建议。

Subcommands

| **Subcommand** | **用法** |
| --- | --- |
| ls | 列举所有复制区域 |
| get | 查看YAML文件关于复制区域的内容 |
| set | 创建或修改复制区域 |
| rm | 移除一个复制区域 |

术语

*# List all replication zones:*

cockroach zone ls <flags>

*# View the default replication zone for the cluster:*

cockroach zone get .default <flags>

*# View the replication zone for a database:*

cockroach zone get <database> <flags>

*# View the replication zone for a table:*

cockroach zone get <database.table> <flags>

*# Edit the default replication zone for the cluster:*

cockroach zone set .default --file=<zone-content.yaml> <flags>

*# Create/edit the replication zone for a database:*

cockroach zone set <database> --file=<zone-conent.yaml> <flags>

*# Create/edit the replication zone for a table:*

cockroach zone set <database.table> --file=<zone-content.yaml> <flags>

*# Create/edit the replication zone for a table partition:*

cockroach zone set <database.table.partition> --file=<zone-content.yaml> <flags>

*# Remove the replication zone for a database:*

cockroach zone rm <database> <flags>

*# Remove the replication zone for a table:*

cockroach zone rm <database.table> <flags>

*# View help:*

cockroach zone --help

cockroach zone ls --help

cockroach zone get --help

cockroach zone set --help

cockroach zone rm --help

Flags

通用

| **Flag** | **简介** |
| --- | --- |
| --disable-replication | 通过设置区域冗余副本数量为1，关闭区域数据冗余的功能。 |
| --echo-sql | （New in v1.1）显示命令行工具成功发送的SQL语句。 |
| --file -f | 定义复制区域的YAML文件位置。用户若需要通过标准输入指定复制区域的配置，可以使用-作为该Flag的值。 该Flag只能配置set子命令使用。 |

客户端连接

| **Flag** | **简介** |
| --- | --- |
| --host | 指定服务器ip地址，可以是集群的任意节点。  **环境变量**：COCKROACH\_HOST **默认**：localhost |
| --port -p | 指定服务器端口  **环境变量**：COCKROACH\_PORT **默认**：26257 |
| --user -u | 指定[SQL用户](http://doc.cockroachchina.baidu.com/#deploy/access-management/manage-users/)  **环境变量**：COCKROACH\_USER **默认**：root |
| --insecure | 使用非安全模式的连接  **环境变量**：COCKROACH\_INSECURE **默认**：false |
| --certs-dir | 指定存放着[CA证书和客户端证书密钥的目录](http://doc.cockroachchina.baidu.com/#deploy/create-security-certificates/)  **环境变量**：COCKROACH\_CERTS\_DIR **默认**：${HOME}/.cockroach-certs/ |
| --url | 指定[connection url](http://doc.cockroachchina.baidu.com/#develop/client-connection-parameters/#connect-using-a-url)  **环境变量**：COCKROACH\_URL **默认**：空 |

更多细节可以查看：[客户端连接参数](http://doc.cockroachchina.baidu.com/#develop/client-connection-parameters/)

目前，只有root账户能够配置复制区域。

日志

cockroach zone命令默认输出错误日志到stderr，详见[日志行为](https://www.cockroachlabs.com/docs/v2.0/debug-and-error-logs)

基本示例

以下示例侧重于配置复制区域的基本方法和语法。

列举预配置的复制区域

新创建CockroachDB集群使用预配置的复制区域：

cockroach zone ls --insecure

*# 输出：*

.default

.liveness

.meta

system.jobs

查看默认的复制区域

覆盖集群的复制区域（.default）默认情况下将数据冗余到集群的任意3个节点，当Range副本在超过67108864 bytes时副本将分裂。

查看默认的复制区域，可执行cockroach zone get .default：

cockroach zone get .default --insecure

*# 输出:*

.default

range\_min\_bytes: 1048576

range\_max\_bytes: 67108864

gc:

ttlseconds: 86400

num\_replicas: 3

constraints: []

编辑默认的复制区域

**WARNING:** 修改.default复制区域不会应用到其他已有的复制区域。如果用户需要增加.default的冗余副本数，用户还需要增加[必要内部数据](http://doc.cockroachchina.baidu.com/#deploy/configure-replication-zones/#create-a-replication-zone-for-a-system-range)的冗余副本数。 用户若需要编辑默认复制区域，首先创建仅定义了需要更改的内容的YAML文件（其他内容将延用.default区域），然后执行cockroach zone set .default -f <file.yaml>命令：

cat default\_update.yaml

num\_replicas: 5

cockroach zone set .default --insecure -f default\_update.yaml

*# 输出：*

range\_min\_bytes: 1048576

range\_max\_bytes: 67108864

gc:

ttlseconds: 86400

num\_replicas: 5

constraints: []

此外，用户可以通过标准输入配置YAML内容：

echo 'num\_replicas: 5' | cockroach zone set .default --insecure -f -

为数据库创建复制区域

用户若需要为特定的数据库配置复制区域，首先创建仅定义了需要更改的内容的YAML文件（其他内容将不发生变更），然后执行cockroach zone set <database> -f <file.yaml>命令：

cat database\_zone.yaml

num\_replicas: 7

cockroach zone set db1 --insecure -f database\_zone.yaml

*# 输出*

range\_min\_bytes: 1048576

range\_max\_bytes: 67108864

gc:

ttlseconds: 86400

num\_replicas: 5

constraints: []

此外，用户可以通过标准输入配置YAML内容：

echo 'num\_replicas: 5' | cockroach zone set db1 --insecure -f -

为表创建复制区域

用户若需要为特定的表配置复制区域，首先创建仅定义了需要更改的内容的YAML文件（其他内容不发生变更），然后执行cockroach zone set <database.table> -f <file.yaml>命令：

cat table\_zone.yaml

num\_replicas: 7

cockroach zone set db1.t1 --insecure -f table\_zone.yaml

*# 输出*

range\_min\_bytes: 1048576

range\_max\_bytes: 67108864

gc:

ttlseconds: 86400

num\_replicas: 7

constraints: []

此外，用户可以通过标准输入配置YAML内容：

echo 'num\_replicas: 7' | cockroach zone set db1.t1 --insecure -f -

为分区创建复制区域

（New in v2.0）

**NOTE:** 此为企业版功能。

为了控制分区的数据冗余复制，用户需要创建仅定义需要更改的内容的YAML文件（其他内容不发生变更），然后执行cockroach zone set <database.table.partition> -f <file.yaml>命令：

cat > australia.zone.yml

constraints: [+datacenter=au1]

此外，用户使用cockroach zone命令配置特定的区域：

cockroach zone set roachlearn.students\_by\_list.australia --insecure -f australia.zone.yml

为系统Range创建复制区域

除了SQL接口可查询到的数据库和表之外，CockroachDB在系统range还存储了一些内部数据。CockroachDB为这些Range预先配置了复制区域：

| **区域名** | **简介** |
| --- | --- |
| .meta | "meta" Range包含集群所有数据的位置信息。  考虑到在"meta"数据上不会进行历史查询，且为获得可靠性"mata"数据的Range较小，CockroachDB**预先配置**.meta复制区域，使得这些Range的ttlseconds值低于默认值。  若集群是多数据中心部署，建议用户在每个数据中心配置一个meta ranges的副本。 |
| .liveness | "liveness" Range包含集群任意时间点下存活着的节点的信息。  如同"meta"数据，该Range数据不会进行历史查询。因此CockroachDB会**预先配置**.liveness复制区域，使得这些Range的ttlseconds值低于默认值。  假设该Range不可用，集群将不可用，因此推荐配置该Range更高的冗余副本数。 |
| .timeseries | "timeseries" Range包含有关集群的监视数据，以支持CockroachDB的Admin界面中的图形展示。如有必要，用户可以自定义.timeseries的复制区域。 |
| .system | 对应其他一些重要的内部数据，包括创建新表ID或跟踪集群节点所需要的信息。如有必要，用户可以自定义.system的复制区域。 |

为了控制系统Ranges的数据冗余复制，用户需要创建仅定义需要更改的内容的YAML文件（其他内容不发生变更），然后执行cockroach zone set <zone-name> -f <file.yaml>命令：

cat meta\_zone.yaml

num\_replicas: 7

cockroach zone set .meta --insecure -f meta\_zone.yaml

*# 输出*

range\_min\_bytes: 1048576

range\_max\_bytes: 67108864

gc:

ttlseconds: 86400

num\_replicas: 7

constraints: []

此外，用户可以通过标准输入配置YAML内容：

echo 'num\_replicas: 7' | cockroach zone set .meta --insecure -f -

输出命令行工具成功发送的SQL语句

使用--echo-sqlFlag显示命令行工具成功发送的SQL语句：

echo 'num\_replicas: 5' | cockroach zone set .default --insecure --echo-sql -f -

> BEGIN

> SAVEPOINT cockroach\_restart

> SELECT config FROM system.zones WHERE id = $1

> UPSERT INTO system.zones (id, config) VALUES ($1, $2)

range\_min\_bytes: 1048576

range\_max\_bytes: 67108864

gc:

ttlseconds: 90000

num\_replicas: 5

constraints: []

> RELEASE SAVEPOINT cockroach\_restart

> COMMIT

基于场景的示例

跨数据中心的数据冗余

场景描述

* 集群部署在3个数据中心，共有6个节点，每个数据中心2个节点。
* 数据冗余3份副本，均衡分布在3个数据中心。

方式

使用--localityFlag启动每个节点：

*# Start the two nodes in datacenter 1:*

cockroach start --insecure --host=<node1 hostname> --locality=datacenter=us-1

cockroach start --insecure --host=<node2 hostname> --locality=datacenter=us-1 \

--join=<node1 hostname>:27257

*# Start the two nodes in datacenter 2:*

cockroach start --insecure --host=<node3 hostname> --locality=datacenter=us-2 \

--join=<node1 hostname>:27257

cockroach start --insecure --host=<node4 hostname> --locality=datacenter=us-2 \

--join=<node1 hostname>:27257

*# Start the two nodes in datacenter 3:*

cockroach start --insecure --host=<node5 hostname> --locality=datacenter=us-3 \

--join=<node1 hostname>:27257

cockroach start --insecure --host=<node6 hostname> --locality=datacenter=us-3 \

--join=<node1 hostname>:27257

不需更改复制区域的配置。默认情况下，集群进行3份数据冗余。在没有明确的约束的情况下，集群会尽可能地使节点本地的数据副本多样化。

针对特定数据中心进行单个副本约束

场景描述

* 集群部署在5个数据中心，共有5个节点，每个数据中心1个节点。
* 数据冗余3份。其中西海岸的数据根据法律规定只能存储在西海岸的节点上，而全国的数据则散布在全国各个节点上。

方式

* 使用--localityFlag启动每个节点：

*# Start the four nodes:*

cockroach start --insecure --host=<node1 hostname> --locality=region=us-west1,datacenter=us-west1-a

cockroach start --insecure --host=<node2 hostname> --locality=region=us-west1,datacenter=us-west1-b \

--join=<node1 hostname>:27257

cockroach start --insecure --host=<node3 hostname> --locality=region=us-central1,datacenter=us-central1-a \

--join=<node1 hostname>:27257

cockroach start --insecure --host=<node4 hostname> --locality=region=us-east1,datacenter=us-east1-a \

--join=<node1 hostname>:27257

cockroach start --insecure --host=<node4 hostname> --locality=region=us-east1,datacenter=us-east1-b \

--join=<node1 hostname>:27257

* 配置西海岸数据库的复制区域

*# Create a YAML file with the replica count set to 5:*

cat west\_app\_zone.yaml

constraints: {"+region=us-west1": 2, "+region=us-central1": 1}

*# Apply the replication zone to the database used by the West Coast application:*

cockroach zone set west\_app\_db --insecure -f west\_app\_zone.yaml

range\_min\_bytes: 1048576

range\_max\_bytes: 67108864

gc:

ttlseconds: 86400

num\_replicas: 3

constraints: {+region=us-central1: 1, +region=us-west1: 2}

数据库三份数据冗余副本中的两个将放在region=us-west1中，剩下的将放在region=us-central1中。该部署方式为应用程序提供了在任何一个数据中心完全故障的情况下恢复数据库业务能力，同时由于西海岸拥有大多数副本，则该区域的读写延迟低。

* 其他数据库无需额外的配置。

集群配置默认为数据冗余3份，并尽可能均衡地散布副本。因为在每个节点的--locality中指定的第一个键值对被认为是位置序列的最重要部分，所以CockroachDB会在三个不同区域中各放置一个副本。

多应用读写各自数据库

场景描述

* 有两个独立的应用程序，连接同一个CockroachDB集群中单独的数据库。
* 集群部署在2个数据中心，共6个节点，每个数据中心3个节点。
* 需要将应用1的数据冗余5份，均衡分布到2个数据中心的节点。
* 需要将应用2的数据冗余3份，只散布到其中一个数据中心。

方式

* 使用--localityFlag启动每个节点：

*# Start the three nodes in datacenter 1:*

cockroach start --insecure --host=<node1 hostname> --locality=datacenter=us-1

cockroach start --insecure --host=<node2 hostname> --locality=datacenter=us-1 \

--join=<node1 hostname>:27257

cockroach start --insecure --host=<node3 hostname> --locality=datacenter=us-1 \

--join=<node1 hostname>:27257

*# Start the three nodes in datacenter 2:*

cockroach start --insecure --host=<node4 hostname> --locality=datacenter=us-2 \

--join=<node1 hostname>:27257

cockroach start --insecure --host=<node5 hostname> --locality=datacenter=us-2 \

--join=<node1 hostname>:27257

cockroach start --insecure --host=<node6 hostname> --locality=datacenter=us-2 \

--join=<node1 hostname>:27257

* 配置应用1对应数据库的复制空间

*# Create a YAML file with the replica count set to 5:*

cat app1\_zone.yaml

num\_replicas: 5

*# Apply the replication zone to the database used by application 1:*

cockroach zone set app1\_db --insecure -f app1\_zone.yaml

range\_min\_bytes: 1048576

range\_max\_bytes: 67108864

gc:

ttlseconds: 86400

num\_replicas: 5

constraints: []

应用1的数据不需要任何配置，集群将平衡数据中心1和2之间的应用1使用的数据库中的数据。

* 配置应用2对应数据库的复制空间

*# Create a YAML file with 1 datacenter as a required constraint:*

cat app2\_zone.yaml

constraints: [+datacenter=us-2]

*# Apply the replication zone to the database used by application 2:*

cockroach zone set app2\_db --insecure -f app2\_zone.yaml

range\_min\_bytes: 1048576

range\_max\_bytes: 67108864

gc:

ttlseconds: 86400

num\_replicas: 3

constraints: [+datacenter=us-2]

添加了针对应用2的数据仅在us-2数据中心内复制的约束。

针对特定表配置更严格的复制

场景描述

* 集群共有7个节点，其中5个配备SSD硬盘，2个配备HDD硬盘。
* 数据默认情况下冗余3份。
* 存在特定表的数据重要程度高且使用频繁，需要指定其冗余5份副本并散布到配有SSD硬盘上节点上。

方式

* 启动每个节点，指定--store属性为ssd或hdd：

*# Start the 5 nodes with SSD storage:*

cockroach start --insecure --host=<node1 hostname> --store=path=node1,attrs=ssd

cockroach start --insecure --host=<node2 hostname> --store=path=node2,attrs=ssd \

--join=<node1 hostname>:27257

cockroach start --insecure --host=<node3 hostname> --store=path=node3,attrs=ssd \

--join=<node1 hostname>:27257

cockroach start --insecure --host=<node4 hostname> --store=path=node4,attrs=ssd \

--join=<node1 hostname>:27257

cockroach start --insecure --host=<node5 hostname> --store=path=node5,attrs=ssd \

--join=<node1 hostname>:27257

*# Start the 2 nodes with HDD storage:*

cockroach start --insecure --host=<node6 hostname> --store=path=node6,attrs=hdd \

--join=<node1 hostname>:27257

cockroach start --insecure --host=<node7 hostname> --store=path=node7,attrs=hdd \

--join=<node1 hostname>:27257

* 针对特定表配置更严格复制策略的复制空间

*# Create a YAML file with the replica count set to 5*

*# and the ssd attribute as a required constraint:*

cat table\_zone.yaml

num\_replicas: 5

constraints: [+ssd]

*# Apply the replication zone to the table:*

cockroach zone set db.important\_table --insecure -f table\_zone.yaml

range\_min\_bytes: 1048576

range\_max\_bytes: 67108864

gc:

ttlseconds: 86400

num\_replicas: 5

constraints: [+ssd]

数据库表important\_table的数据将冗余5份，并按照复制约束将副本存储在ssd硬盘上。

调整系统Range复制

场景

* 集群部署在7个数据中心
* 数据默认冗余5份
* 为提高性能，在每个数据中心复制一份meta range
* 为节省存储空间，内部时间序列数据只冗余3份

方式

* 使用--localityFlag启动每个节点：

cockroach start --insecure --host=<node1 hostname> --locality=datacenter=us-1

cockroach start --insecure --host=<node2 hostname> --locality=datacenter=us-2 \

--join=<node1 hostname>:27257

cockroach start --insecure --host=<node3 hostname> --locality=datacenter=us-3 \

--join=<node1 hostname>:27257

cockroach start --insecure --host=<node4 hostname> --locality=datacenter=us-4 \

--join=<node1 hostname>:27257

cockroach start --insecure --host=<node5 hostname> --locality=datacenter=us-5 \

--join=<node1 hostname>:27257

cockroach start --insecure --host=<node6 hostname> --locality=datacenter=us-6 \

--join=<node1 hostname>:27257

cockroach start --insecure --host=<node7 hostname> --locality=datacenter=us-7 \

--join=<node1 hostname>:27257

* 选择任意节点，配置.default复制空间

echo 'num\_replicas: 5' | cockroach zone set .default --insecure -f -

range\_min\_bytes: 1048576

range\_max\_bytes: 67108864

gc:

ttlseconds: 86400

num\_replicas: 5

constraints: []

此时集群的所有数据，包括应用数据和内部系统数据，将冗余5份副本。

* 选择任意节点，配置.meta复制空间

echo 'num\_replicas: 7' | cockroach zone set .meta --insecure -f -

range\_min\_bytes: 1048576

range\_max\_bytes: 67108864

gc:

ttlseconds: 86400

num\_replicas: 7

constraints: []

此时集群的.meta数据将冗余7份，每个数据中心一个数据副本，而其他数据将冗余5份副本。

* 选择任意节点，配置.timeseries复制空间

echo 'num\_replicas: 3' | cockroach zone set .timeseries --insecure -f -

range\_min\_bytes: 1048576

range\_max\_bytes: 67108864

gc:

ttlseconds: 86400

num\_replicas: 3

constraints: []

此时集群的.timeseries数据将冗余3份，其他数据复制策略不变。