通过本小节你将了解到CockroachDB如何在出现错误以及恢复的过程中保持系统集群的稳定可用，下面将模拟节点丢失以及节点补充的过程。

Step 1: 启动3个节点

开启新终端，执行如下命令添加节点1

cockroach start --insecure --store=fault-node1 --host=localhost --port=26257 --http-port=8080 --join=localhost:26257,localhost:26258,localhost:26259

开启新终端，执行如下命令添加节点2

cockroach start --insecure --store=fault-node2 --host=localhost --port=26258 --http-port=8081 --join=localhost:26257,localhost:26258,localhost:26259

开启新终端，执行如下命令添加节点3

cockroach start --insecure --store=fault-node3 --host=localhost --port=26259 --http-port=8082 --join=localhost:26257,localhost:26258,localhost:26259

Step 2: 初始化集群

cockroach init --insecure --host=localhost --port=26257

Step 3: 验证集群的可用性

连接任意一个节点，尝试SQL查询。

启动交互式shell：

cockroach sql --insecure --port=26257

查询：

**SHOW** **DATABASES**;

+*--------------------+*

| Database |

+*--------------------+*

| system |

+*--------------------+*

(1 row)

Step 4: 移除节点2，模拟节点丢失

cockroach quit --insecure --port=26258

initiating graceful shutdown of server

ok

Step 5: 验证剩余节点的可用性

选择节点2或是节点3，尝试SQL查询。

启动交互式shell：

cockroach sql --insecure --port=26259

查询：

**SHOW** **DATABASES**;

+*--------------------+*

| Database |

+*--------------------+*

| bank |

| system |

+*--------------------+*

(2 rows)

可见集群当中的绝大多数副本（此情况下是2/3）存活，则系统保持可用状态。假设集群只剩下一个节点，最后一个节点将停止响应操作直至集群再次恢复（或是新增）一个可用节点，系统将恢复绝大多数副本存活的可用状态。

Step 6: 尝试在丢失节点的情况下写数据

数据生成

选择连接节点2或是节点3，使用cockroach gen工具生成新的数据库startrek：

cockroach gen example-data startrek | cockroach sql --insecure

CREATE DATABASE

SET

DROP TABLE

DROP TABLE

CREATE TABLE

INSERT 79

CREATE TABLE

INSERT 200

数据验证

选择连接节点1（port 26257）或节点3（port 26259），验证数据是否成功生成。

进入交互式shell：

cockroach sql --insecure --port=26259

查询：

**SHOW** **DATABASES**;

+*--------------------+*

| Database |

+*--------------------+*

| startrek |

| system |

+*--------------------+*

(2 rows)

**SHOW** **TABLES** **FROM** startrek;

+*----------+*

| Table |

+*----------+*

| episodes |

| quotes |

+*----------+*

(2 rows)

**SELECT** \* **FROM** startrek.episodes **LIMIT** 10;

+*----+--------+-----+--------------------------------+----------+*

| id | season | num | title | stardate |

+*----+--------+-----+--------------------------------+----------+*

| 1 | 1 | 1 | The Man Trap | 1531.1 |

| 2 | 1 | 2 | Charlie X | 1533.6 |

| 3 | 1 | 3 | Where No Man Has Gone Before | 1312.4 |

| 4 | 1 | 4 | The Naked Time | 1704.2 |

| 5 | 1 | 5 | The Enemy Within | 1672.1 |

| 6 | 1 | 6 | Mudd's Women | 1329.8 |

| 7 | 1 | 7 | What Are Little Girls Made Of? | 2712.4 |

| 8 | 1 | 8 | Miri | 2713.5 |

| 9 | 1 | 9 | Dagger of the Mind | 2715.1 |

| 10 | 1 | 10 | The Corbomite Maneuver | 1512.2 |

+*----+--------+-----+--------------------------------+----------+*

(10 rows)

Step 7: 恢复节点2

cockroach **start** *--insecure --store=fault-node2 --host=localhost --port=26258 --http-port=8081 --join=localhost:26257*

CockroachDB node **starting** **at** 2018-08-13 15:10:52.34274101 +0000 UTC

build: CCL v2.0.5 @ 2018/08/13 14:48:26 (go1.10)

admin: http://localhost:8081

**sql**: postgresql://root@localhost:26258?sslmode=disable

**logs**: node2/**logs**

store[0]: path=fault-node2

**status**: restarted pre-existing node

clusterID: {5638ba53-fb77-4424-ada9-8a23fbce0ae9}

nodeID: 2

Step 8: 验证新增节点2是否同步增量数据

启动交互式shell：

cockroach sql --insecure --port=26258

查询：

**SELECT** \* **FROM** startrek.episodes **LIMIT** 10;

+*----+--------+-----+--------------------------------+----------+*

| id | season | num | title | stardate |

+*----+--------+-----+--------------------------------+----------+*

| 1 | 1 | 1 | The Man Trap | 1531.1 |

| 2 | 1 | 2 | Charlie X | 1533.6 |

| 3 | 1 | 3 | Where No Man Has Gone Before | 1312.4 |

| 4 | 1 | 4 | The Naked Time | 1704.2 |

| 5 | 1 | 5 | The Enemy Within | 1672.1 |

| 6 | 1 | 6 | Mudd's Women | 1329.8 |

| 7 | 1 | 7 | What Are Little Girls Made Of? | 2712.4 |

| 8 | 1 | 8 | Miri | 2713.5 |

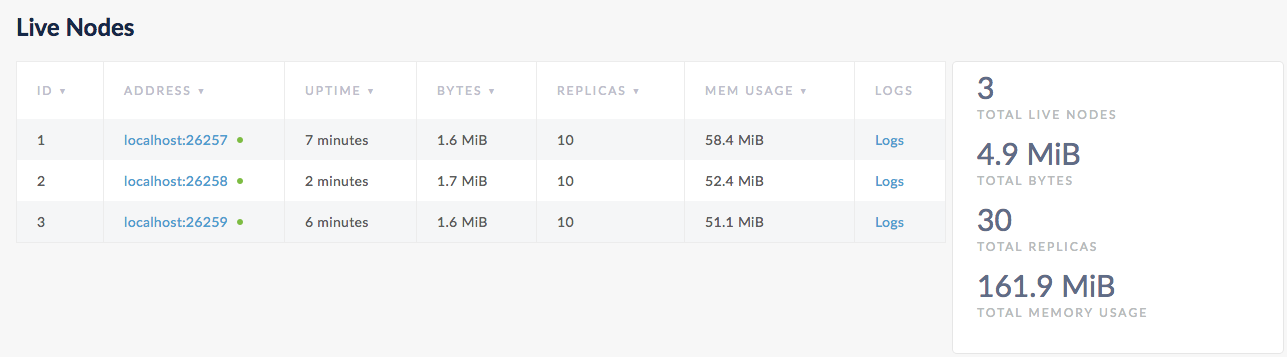
| 9 | 1 | 9 | Dagger of the Mind | 2715.1 |

| 10 | 1 | 10 | The Corbomite Maneuver | 1512.2 |

+*----+--------+-----+--------------------------------+----------+*

(10 rows)

可见增量数据自动同步完成。 也可登陆Admin界面（http://localhost:8080）查看节点信息：



**TIPS:** CockroachDB默认冗余副本数为3，可通过cockroach zone指令调整，详见：[复制区域](http://doc.cockroachchina.baidu.com/#deploy/configure-replication-zones)

Step 9: 新增节点4

开启新的终端，执行：

cockroach start --insecure --store=fault-node4 --host=localhost --port=26260 --http-port=8083 --join=localhost:26257,localhost:26258,localhost:26259

CockroachDB node starting at 2018-08-13 15:10:52.34274101 +0000 UTC

build: CCL v2.0.5 @ 2018/08/13 14:48:26 (go1.10)

admin: http://localhost:8083

sql: postgresql://root@localhost:26260?sslmode=disable

logs: node4/logs

store[0]: path=fault-node4

status: initialized new node, joined pre-existing cluster

clusterID: {5638ba53-fb77-4424-ada9-8a23fbce0ae9}

nodeID: 4

Step 10: 移除节点2

在节点2终端使用Ctrl+C终止程序，或在新终端执行：

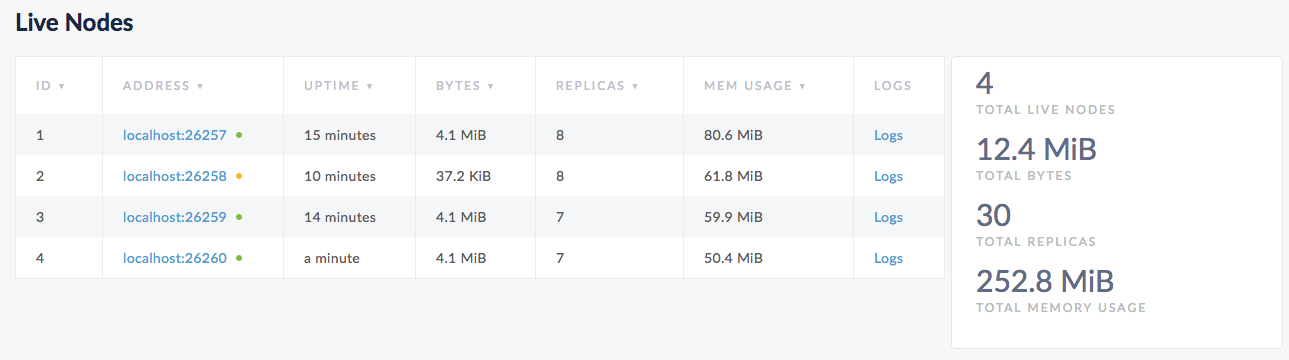
cockroach quit --insecure --port=26258

initiating graceful shutdown of server

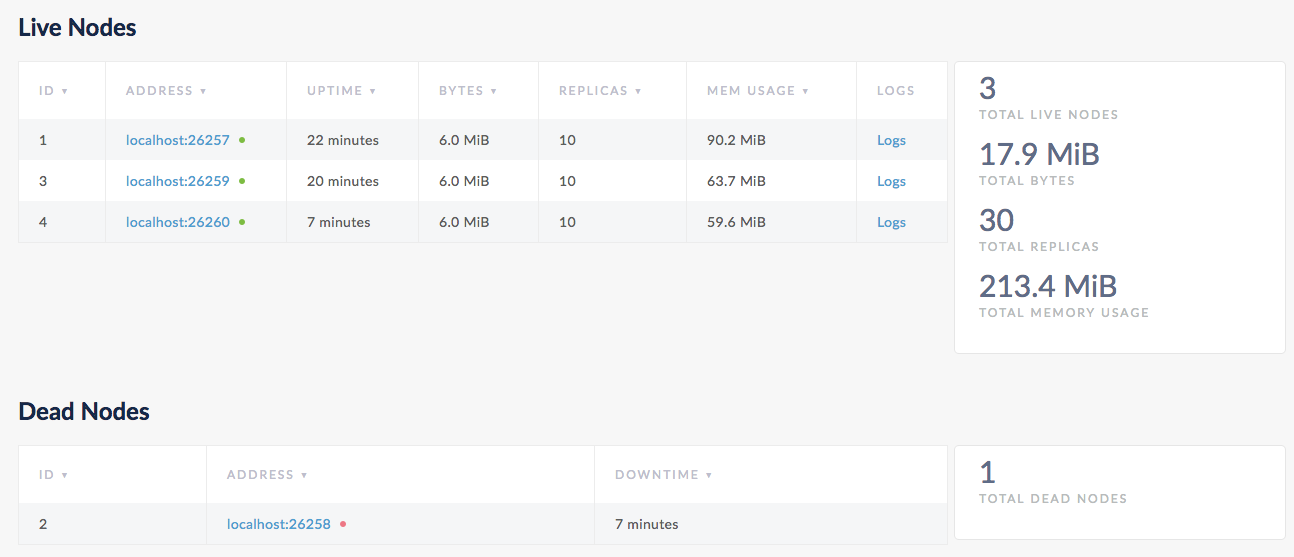
ok

server drained and shutdown completed

Step 11: 查看剩余节点的数据冗余情况

登陆Admin界面（http://localhost:8080）查看节点信息： 

当前可见存活节点有4个。 大约1min过后，由于节点2一直未响应请求，将被标黄。



大约10min过后，节点2将从**Live Nodes**标签页转移到**Dead Nodes**标签页，集群不再期待节点2的恢复。 此时可以看见集群当中的冗余副本重新均分到剩下的存活节点。

Step 12: 关闭集群

在各个终端使用Ctrl+C终止程序，如不再需要测试用的集群，可使用如下指令删除：

rm -rf fault-node1 fault-node2 fault-node3 fault-node4 fault-node5

**NOTE:** 对于最后两个节点，可能需要花大约1min左右的时间去终止程序。原因是最后只剩下2个节点在集群当中的时候，由于已经丧失了集群当中冗余副本的绝大多数（此情况下是3/5），集群不再允许操作。如需加速终止程序的进程，可以尝试多次Ctrl+C操作。