cockroach gen命令能够生成命令行界面工具（例如[man手册](https://en.wikipedia.org/wiki/Man_page)和自动补全bash脚本），能够生成样本数据添加到测试数据库当中，也能够生成集群负载均衡用的HAProxy配置文件。

概要

*# Generate man pages:*

cockroach gen man

*# Generate bash autocompletion script:*

cockroach gen autocomplete

*# Generate example SQL data:*

cockroach gen example-data intro | cockroach sql

cockroach gen example-data startrek | cockroach sql

*# Generate an HAProxy config file for a running cluster:*

cockroach gen haproxy

*# View help:*

cockroach gen --help

cockroach gen man --help

cockroach gen autocomplete --help

cockroach gen example-data --help

cockroach gen haproxy --help

子命令

| **子命令** | **简介** |
| --- | --- |
| man | 生成CockroachDB man手册 |
| autocomplete | 生成CockroachDB自动补全脚本 |
| example-data | 生成样本数据 |
| haproxy | 生成Cockroach集群负载均衡用的HAProxy配置文件 |

Flags

通用

子命令man

| **Flag** | **简介** |
| --- | --- |
| --path | 生成的man手册存放的路径  **默认:** 当前目录下的man/man1子目录 |

子命令autocomplete

| **Flag** | **简介** |
| --- | --- |
| --out | 生成的自动补全脚本存放的路径  **默认:** 当前目录下的cockroach.bash子目录 |

子命令example-data

无对应的Flag，详见[生成测试数据](https://www.cockroachlabs.com/docs/stable/generate-cockroachdb-resources.html#generate-example-data)

子命令haproxy

| **Flag** | **简介** |
| --- | --- |
| --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/ |
| --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 |
| --insecure | 使用非安全模式的连接，否则需要使用--certs-dirFlag指定有效的证书目录  **环境变量**：COCKROACH\_INSECURE **默认**：false |
| --out | haproxy.cfg文件存放的路径。若同名文件已存在，会进行覆写。  **默认**：当前目录 |
| --port -p | 指定服务器端口  **环境变量**：COCKROACH\_PORT **默认**：26257 |

日志

cockroach gen默认输出日志错误到stderr，更丰富的日志功能可查看：[日志行为](https://www.cockroachlabs.com/docs/stable/debug-and-error-logs.html)

示例

生成man手册

*# Generate man pages:*

cockroach gen man

*# Move the man pages to the man directory:*

sudo mv man/man1/\* /usr/share/man/man1

*# Access man pages:*

man cockroach

生成自动补全bash脚本

*# Generate bash autocompletion script:*

cockroach gen autocomplete

*# Add the script to your .bashrc and .bash\_profle:*

printf "\n\n#cockroach bash autocomplete\nsource '<path to>cockroach.bash'" >> ~/.bashrc

printf "\n\n#cockroach bash autocomplete\nsource '<path to>cockroach.bash'" >> ~/.bash\_profile

通过该脚本用户可以使用tab自动补全cockroach命令。

生成测试数据

生成startrek数据库

该数据库包含了episodes和quotes两张表。

*# Generate example `startrek` database:*

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

CREATE DATABASE

SET

DROP TABLE

DROP TABLE

CREATE TABLE

INSERT 79

CREATE TABLE

INSERT 200

启动内置SQL客户端：

*# Launch the built-in SQL client to view it:*

cockroach sql --insecure

执行SQL语句：

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

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

| Table |

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

| episodes |

| quotes |

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

(2 rows)

生成intro数据库

该数据库包含了mytable一张表。

*# Generate example `intro` database:*

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

CREATE DATABASE

SET

DROP TABLE

CREATE TABLE

INSERT 1

INSERT 1

INSERT 1

INSERT 1

...

启动内置SQL客户端：

*# Launch the built-in SQL client to view it:*

cockroach sql --insecure

执行SQL语句：

**SHOW** **TABLES** **FROM** intro;

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

| Table |

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

| mytable |

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

(1 row)

**SELECT** \* **FROM** intro.mytable **WHERE** (l % 2) = 0;

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

| l | v |

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

| 0 | !\_\_aaawwmqmqmwwwaas,,\_ .\_\_aaawwwmqmqmwwaaa,, |

| 2 | !"VT?!"""^~~^"""??T$Wmqaa,\_auqmWBT?!"""^~~^^""??YV^ |

| 4 | ! "?##mW##?"- |

| 6 | ! C O N G R A T S \_am#Z??A#ma, Y |

| 8 | ! \_ummY" "9#ma, A |

| 10 | ! vm#Z( )Xmms Y |

| 12 | ! .j####mmm#####mm#m##6. |

| 14 | ! W O W ! jmm###mm######m#mmm##6 |

| 16 | ! ]#me\*Xm#m#mm##m#m##SX##c |

| 18 | ! dm#||+\*$##m#mm#m#Svvn##m |

| 20 | ! :mmE=|+||S##m##m#1nvnnX##; A |

| 22 | ! :m#h+|+++=Xmm#m#1nvnnvdmm; M |

| 24 | ! Y $#m>+|+|||##m#1nvnnnnmm# A |

| 26 | ! O ]##z+|+|+|3#mEnnnnvnd##f Z |

| 28 | ! U D 4##c|+|+|]m#kvnvnno##P E |

| 30 | ! I 4#ma+|++]mmhvnnvq##P` ! |

| 32 | ! D I ?$#q%+|dmmmvnnm##! |

| 34 | ! T -4##wu#mm#pw##7' |

| 36 | ! -?$##m####Y' |

| 38 | ! !! "Y##Y"- |

| 40 | ! |

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

(21 rows)

生成HAProxy配置文件

[HAProxy](http://www.haproxy.org/)是一款比较流行的开源TCP负载均衡器。CockroachDB内置了相应的命令，用于生成集群的HAProxy配置文件。

执行cockroach gen haproxy命令生成配置文件：

#非安全模式

cockroach gen haproxy --insecure \

--host=<address of any node **in** the cluster> \

--port=26257

#安全模式

cockroach gen haproxy \

--certs-dir=<path to certs directory> \

--host=<address of any node **in** the cluster> \

--port=26257

该命令生成了haproxy.cfg配置文件，预设了当前集群的最低限度的[配置](http://cbonte.github.io/haproxy-dconv/1.7/configuration.html)，内容如下：

global

maxconn 4096

defaults

mode tcp

*# Timeout values should be configured for your specific use.*

*# See: https://cbonte.github.io/haproxy-dconv/1.8/configuration.html#4-timeout%20connect*

timeout connect 10s

timeout client 1m

timeout server 1m

*# TCP keep-alive on client side. Server already enables them.*

option clitcpka

listen psql

bind :26257

mode tcp

balance roundrobin

option httpchk GET /health?ready=1

server cockroach1 <node1 address>:26257

server cockroach2 <node2 address>:26257

server cockroach3 <node3 address>:26257

涉及到配置项：

| **配置项** | **简介** |
| --- | --- |
| timeout connect timeout client timeout server | 集群各部分timeout的时间值 |
| bind | HAProxy的监听端口，注意修改系统相关网络配置，确保该端口能够开启并访问。  推荐将HAProxy部署到CockroachDB集群以外的单独机器上，若部署在集群节点上（不推荐），需要注意26257端口占用问题。 |
| balance | 均衡算法，此处为roundrobin算法，以简单的轮询方式将连接均衡到各个节点，更多算法详见:[HAProxy Configuration Manual](http://cbonte.github.io/haproxy-dconv/1.7/configuration.html#4-balance) |
| option httpchk | HTTP端点供HAProxy用来检查节点的健康状况，此处[/health?ready=1](http://doc.cockroachchina.baidu.com/#deploy/monitoring-and-alerting/overview/#health-ready-1)能够保证HAProxy不会将请求转发到存活但未准备好接受请求的节点上。 |
| server | 该配置指定集群当中节点的地址（例如节点启动时--host指定的地址）和用于确保各节点健康的监听端口。 |

**NOTE:** 更多介绍和配置，详见：[HAProxy Configuration Manual](http://cbonte.github.io/haproxy-dconv/1.7/configuration.html#4-balance)