cockroachdb基本命令

测试上的节点：

172.18.44.120:26257 8080

172.18.44.120:26258 8081

172.18.44.128:26257 8080

172.18.44.128:26258 8081

172.18.44.123:26257 8080

172.18.44.123:26258 8081

测试机器密码

huaxun!@#

## 1.启动节点:

（安全）

|  |
| --- |
| cockroach start --certs-dir=certs --background \  --store=node2 \  --host=172.18.44.128 \  --port=26260 \  --http-port=8088 \  --cache=10% \  --max-sql-memory=10% \  --join=172.18.44.120:26260 |
| Docker启动:  docker run --name roach1 --net=host \  -e TZ="Asia/Shanghai" \  -v /root/springcloud/CRDB/n1/node1:/cockroach/cockroach-data \  -v /root/springcloud/CRDB/n1/ certs:/cockroach/certs \  -d cockroachdb/cockroach:v2.0.0 \  start --certs-dir=/cockroach/certs \  --host=172.18.44.120 --port=26257 --http-port=8080 --cache=5% --max-sql-memory=5% |

（不安全）:

|  |
| --- |
| cockroach start --insecure --background \  --store=node2 \  --host=172.18.44.120 \  --port=26258 \  --http-port=8081 \  --cache=5% \  --max-sql-memory=5% \  --join=172.18.44.120:26256  java -cp . -Dcom.sun.management.jmxremote.port=8999 -Dcom.sun.managent.jmxremote.authenticate=**false** -Dcom.sun.management.jmxremote.ssl=**false** |
| Docker启动:  docker run --name roach1 --net=host \  -e TZ="Asia/Shanghai" \  -v /root/springcloud/CRDB/n1/node1:/cockroach/cockroach-data \  -v /root/springcloud/CRDB/n1/ certs:/cockroach/certs \  -d cockroachdb/cockroach:v2.0.0 \  start --insecure \  --host=172.18.44.120 --port=26257 --http-port=8080 --cache=5% --max-sql-memory=5% |
|  |

## 2.停止节点：

（安全）

|  |
| --- |
| cockroach quit --certs-dir=certs --port=26257 --host=172.18.44.120 |
| Docker:  **docker exec -it roach1 ./cockroach quit --certs-dir=certs --port=26257 --host=172.18.44.120** |

（不安全）

|  |
| --- |
| cockroach quit --insecure --port=26257 --host=172.18.44.120 |
| Docker:  **docker exec -it roach1 ./cockroach quit --insecure --port=26257 --host=172.18.44.120** |

## 3.删除节点（退役节点）

参考

https://www.cockroachlabs.com/docs/stable/remove-nodes.html#insecure

在你开始之前

确认有足够的节点从要删除的节点接管副本，确定要删除的节点上数据已经迁移。



以下为退役节点（表示将节点的数据迁移到其他节点上，退役成功replicas为0）

（安全）

|  |
| --- |
| cockroach node decommission <node id> --wait=live --certs-dir=certs --host=<address of live node> |
| Docker:  **docker exec -it roach1 ./cockroach node decommission <node id> --wait=live --certs-dir=certs --host=<address of live node>** |

（不安全）

|  |
| --- |
| cockroach node decommission <node id> --insecure --wait=live --host=<address of live node> |
| Docker:  **docker exec -it roach1 ./cockroach node decommission <node id> --insecure --wait=live --host=<address of live node>** |

退役成功（replicas显示为0）才能执行以下命令

|  |
| --- |
| 停用并删除节点（不安全）  cockroach quit --decommission --insecure --host=<address of node to remove> |
| Docker:  **docker exec -it roach1 ./cockroach quit --decommission --insecure --host=<address of node to remove>** |
| 停用并删除节点（安全）  cockroach quit --decommission --certs-dir=certs --host=<address of node to remove> |
| Docker:  **docker exec -it roach1 ./cockroach quit --decommission --certs-dir=certs --host=<address of node to remove>** |

## 4.进入内置sql客户端

（安全）

|  |
| --- |
| cockroach sql --certs-dir=certs --host=<load balancer address>  cockroach sql --certs-dir=certs --host=172.18.44.120 --port=26260 --user=yihongbo |
| Docker:  **docker exec -it roach1 ./cockroach sql --certs-dir=certs --host=172.18.44.120 --port=26260 --user=yihongbo** |

（不安全）

|  |
| --- |
| cockroach sql --insecure --host=<load balancer address> |
| Docker:  **docker exec -it roach1 ./cockroach sql --insecure --host=<load balancer address>** |

## 5.创建和管理用户

注意事项

* 用户名不区分大小写; 必须以字母或下划线开头; 只能包含字母，数字或下划线; 并且必须在1到63个字符之间。
* 创建用户后，您必须[授予他们数据库和表的权限](https://www.cockroachlabs.com/docs/stable/grant.html)。
* 在安全集群上，您必须[为用户创建客户端证书，](https://www.cockroachlabs.com/docs/stable/create-security-certificates.html" \l "create-the-certificate-and-key-pair-for-a-client)并且用户必须[验证他们对集群的访问权限](https://www.cockroachlabs.com/docs/stable/create-and-manage-users.html#user-authentication)。
* 删除用户不会删除该用户的[权限](https://www.cockroachlabs.com/docs/stable/privileges.html)。因此，为了防止未来的用户使用相同的用户名继承旧用户的权限，在删除用户之前或之后[撤销用户的权限](https://www.cockroachlabs.com/docs/stable/revoke.html)是非常重要的。

（1）创建用户

|  |
| --- |
| CREATE USER <username> WITH PASSWORD <password>; 必须root用户才有权限 |

（2）删除用户，在这个例子中，我们首先检查用户的权限。然后，我们撤销用户之前撤销用户的权限

|  |
| --- |
| //查看用户所有权限  SHOW GRANTS ON test.customers FOR<username>;  //移除用户的权限  REVOKE CREATE,INSERT,UPDATE ON test.customers FROM <username>;  //删除用户  DROP USER<username>; |

（3）用户授权

支持的权限

用户可以被授予以下权限。有些权限适用于数据库和表，其他权限仅适用于表（请参见下表中的级别）。

* 当用户被授予对数据库的权限时，在数据库中创建的新表将继承权限，但是权限可以被更改。
* 当一个用户被授予一个表的权限时，权限被限制在该表中。
* 该root用户被自动分配ALL新数据库的权限，并且被允许创建数据库的唯一用户。
* 有关特定语句所需的权限，请参阅相应[SQL语句](https://www.cockroachlabs.com/docs/stable/sql-statements.html)的文档。

| 特权 | 等级 |
| --- | --- |
| ALL | 数据库，表 |
| CREATE | 数据库，表 |
| DROP | 数据库，表 |
| GRANT | 数据库，表 |
| SELECT | 表 |
| INSERT | 表 |
| DELETE | 表 |
| UPDATE | 表 |

1）授予数据库权限

|  |
| --- |
| GRANT CREATE ON DATABASE db1, db2 TO <username>, <username>;  SHOW GRANTS ON DATABASE db1, db2; |

2）授予数据库中特定表的权限

|  |
| --- |
| GRANT DELETE ON TABLE db1.t1, db1.t2 TO <username>;  SHOW GRANTS ON TABLE db1.t1, db1.t2; |

3）授予数据库中所有表的权限

|  |
| --- |
| GRANT SELECT ON TABLE db2.\* TO henryroach;  SHOW GRANTS ON TABLE db2.\*; |

## 6. 为用户创建一个客户端证书和密钥：

|  |
| --- |
| cockroach cert create-client <username> --certs-dir=certs --ca-key=my-safe-directory/ca.key |

## 7. [使用JDBC / JAVA和客户端证书身份验证连接到SSL安全服务器](https://forum.cockroachlabs.com/t/connecting-to-an-ssl-secure-server-using-jdbc-java-and-client-certificate-authentication/400)

## （这不是一个问题，而是我使用JDBC和客户端证书连接到SSL服务器的注意事项）。 另外，客户端证书对于root用户是强制性的（在各种情况下推荐使用）。

给定一个正在运行的受保护的集群，并假定已经创建了一个客户端证书

[https://www.cockroachlabs.com/docs/create-security-certificates.html 25](https://www.cockroachlabs.com/docs/create-security-certificates.html)

使用JDBC进行连接的唯一方法就是postgresql驱动程序不会读取密钥的PEM证书。  
相反，我们会将证书转换为DER格式，并将密钥转换为pks8（比如生成的用户的证书为client.yihongbo.crt，秘钥为client.yihongbo.key）：

|  |
| --- |
| openssl x509 -in client.yihongbo.crt -inform pem -outform der -out yihongbo.der  openssl pkcs8 -topk8 -inform PEM -outform DER -in client.yihongbo.key -out yihongbo.key.pk8 -nocrypt |

连接的url:

|  |
| --- |
| String url = "jdbc:postgresql://172.18.44.120:26257/springcloud?user=yihongbo"  + "&sslcert= yihongbo.der "  + "&sslkey= yihongbo.key.pk8"  + "&sslmode=require&ssl=true"; |

## 8. 数据导入导出

1）SQL转储（导出）详见官网 https://www.cockroachlabs.com/docs/stable/sql-dump.html

|  |
| --- |
| # Dump the schemas and data of specific tables to stdout:  cockroach dump <database> <table> <table...> <flags>  # Dump just the data of specific tables to stdout:  cockroach dump <database> <table> <table...> --dump-mode=data <other flags>  # Dump just the schemas of specific tables to stdout:  cockroach dump <database> <table> <table...> --dump-mode=schema <other flags>  # Dump the schemas and data of all tables in a database to stdout:  cockroach dump <database> <flags>  # Dump just the schemas of all tables in a database to stdout:  cockroach dump <database> --dump-mode=schema <other flags>  # Dump just the data of all tables in a database to stdout:  cockroach dump <database> --dump-mode=data <other flags>  # Dump to a file:  cockroach dump  <database> <table> <flags> > dump-file.sql  # View help:  cockroach dump --help |
|  |

2）数据导入

|  |
| --- |
| cockroach sql --host=172.18.44.120 --port=26257 --insecure --database=springcloud < exampledb.users\_最后一次的最后一次.sql |

{

"secucode": "147123",

"market": "SH",ll

"secuName": null,

"secutype": "0",

"nowprice": 0.0,

"downStopprice": 0.0,

"entrustdifference": 0.0,

"nitiativesellvolume": null,

"status": null,

"entrustbuyqueue": [

{

"price": 0.0,

"volume": 0.0,

"numorders": 0.0

},

{

"price": 0.0,

"volume": 0.0,

"numorders": 0.0

},

{

"price": 0.0,

"volume": 0.0,

"numorders": 0.0

},

{

"price": 0.0,

"volume": 0.0,

"numorders": 0.0

},

{

"price": 0.0,

"volume": 0.0,

"numorders": 0.0

}

],

"entrustsellqueue": [

{

"price": 0.0,

"volume": 0.0,

"numorders": 0.0

},

{

"price": 0.0,

"volume": 0.0,

"numorders": 0.0

},

{

"price": 0.0,

"volume": 0.0,

"numorders": 0.0

},

{

"price": 0.0,

"volume": 0.0,

"numorders": 0.0

},

{

"price": 0.0,

"volume": 0.0,

"numorders": 0.0

}

],

"minData": null,

"tradeDetail": null,

"dateTime": "20180309100953",

"prevClose": 100.0,

"high": 0.0,

"open": 0.0,

"close": 0.0,

"volume": 0.0,

"amount": 0.0,

"low": 0.0,

"entrustRate": 0.0,

"deltaVolume": 0.0,

"swing": 0.0,

"upSpeed": 0.0,

"avgPrice": 0.0,

"changeRate": 0.0,

"upStopPrice": 0.0,

"entrustDifference": null,

"initiativeBuyVolume": 0.0,

"entrustBuyVolume": 0.0,

"entrustSellVolume": 0.0,

"initiativeSellVolume": 0.0

}