Mac 和 Linux

Step 1: 创建桥接网络

用户将在一个物理机上运行多个Docker容器，每个容器上运行一个CockroachDB实例节点。用户需要建立一个桥接网络，使集群中节点能够相互通讯，同时隔离于外部网络。

*#创建了一个名为roachnet的网络*

docker network create -d bridge roachnet

Step 2: 启动节点1

docker run -d --name=roach1 --hostname=roach1 --net=roachnet -p 26257:26257 -p 8080:8080 -v "${PWD}/cockroach-data/roach1:/cockroach/cockroach-data" cockroachdb/cockroach:v2.0.5 start --insecure

上述命令创建了一个容器，并在其中启动了第一个CockroachDB集群节点。 涉及指令和参数的介绍：

* docker run：用Docker命令启动一个容器
* -d： 指定容器运行在后台
* --name：指定容器的名字
* --hostname：指定容器的主机名，和name不同，这个是其他容器加入其所在集群的标签。
* --net：指定Step 1创建的网络名
* -p 26257:26257 -p 8080:8080：指定内部节点和客户端节点的通信端口（26257）映射以及Admin界面从容器到主机的HTTP访问端口映射。
* -v "${PWD}/cockroach-data/roach1:/cockroach/cockroach-data"： 挂载一个主机目录作为容器的磁盘，意味着数据和日志将会存在${PWD}/cockroach-data/roach1。更多细节可查看：[Docker's Bind Mounts](https://docs.docker.com/engine/admin/volumes/bind-mounts/)
* cockroachdb/cockroach:v2.0.5 start --insecure ：启动CockroachDB节点命令

Step 3: 添加节点到集群

现在集群已经运行起来，但是集群只有一个节点，此时用户可以开始连接数据库并进行操作。在实际业务部署中，需要配置大于3个节点才能充分利用CockroachDB的自动冗余副本、数据均衡和集群容错的特性。

添加节点2

docker run -d --name=roach2 --hostname=roach2 --net=roachnet -v "${PWD}/cockroach-data/roach2:/cockroach/cockroach-data" cockroachdb/cockroach:v2.0.5 start --insecure --join=roach1

添加节点3

docker run -d --name=roach3 --hostname=roach3 --net=roachnet -v "${PWD}/cockroach-data/roach3:/cockroach/cockroach-data" cockroachdb/cockroach:v2.0.5 start --insecure --join=roach1

**TIPS:**  
-v：挂载一个主机目录作为容器的磁盘，数据和日志将会存在${PWD}/cockroach-data/roach2或${PWD}/cockroach-data/roach3  
--join：指定节点需要添加到集群的容器的hostname

Step 4: 测试集群

使用docker exec命令启动容器1对应节点1的交互式shell：

docker exec -it roach1 ./cockroach sql --insecure

执行SQL操作：

**CREATE** **DATABASE** bank;

**CREATE** **TABLE** bank.accounts (id INT **PRIMARY** **KEY**, balance DECIMAL);

**INSERT** **INTO** bank.accounts **VALUES** (1, 1000.50);

**SELECT** \* **FROM** bank.accounts;

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

| id | balance |

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

| 1 | 1000.5 |

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

(1 row)

在容器2执行如下命令，启动节点2的交互式shell：

docker exec -it roach2 ./cockroach sql --insecure

查看数据：

**SELECT** \* **FROM** bank.accounts;

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

| id | balance |

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

| 1 | 1000.5 |

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

(1 row)

Step 5: 监控集群

先前已经将容器的8080端口和主机的8080端口做了映射，可以通过http://localhost:8080登陆Admin界面查看集群。

默认情况下，CockroachDB对一份数据自动冗余3份副本到集群当中，均衡分布到各个节点，可在**Replicas per Node**页查看各节点数据冗余情况。

**TIPS:** 关于数据冗余、均衡分布、集群容错的更多介绍，可以查看：[数据复制](http://doc.cockroachchina.baidu.com/#quick-start/explore-benefits/data-replication)、[自动负载均衡](http://doc.cockroachchina.baidu.com/#quick-start/explore-benefits/automatic-rebalancing)、[容错及恢复](http://doc.cockroachchina.baidu.com/#quick-start/explore-benefits/fault-tolerance-and-recovery)

Step 6: 停止集群

使用docker stop和docker rm命令停止和删除容器

docker stop roach1 roach2 roach3

docker rm roach1 roach2 roach3

如不再需要集群数据，可以直接删除容器对应的主机目录

rm -rf cockroach-data

Windows

Step 1: 创建桥接网络

假设用户将在一个物理机上运行多个Docker容器，每个容器上跑一个CockroachDB实例节点。用户需要建立一个桥接网络，使集群中节点能相互通讯，同时隔离于外部网络。

*#创建了一个名为roachnet的网络*

PS C:\Users\username> docker network create -d bridge roachnet

Step 2: 启动节点1

*#注意替换<username>成具体的内容*

PS C:\Users\username> docker run -d `

--name=roach1 `

--hostname=roach1 `

--net=roachnet `

-p 26257:26257 -p 8080:8080 `

-v "//c/Users/<username>/cockroach-data/roach1:/cockroach/cockroach-data" `

cockroachdb/cockroach:v2.0.5 start --insecure

上述命令创建了一个容器，并在其中启动了第一个CockroachDB集群节点。

**TIPS**：  
docker run：用Docker命令启动一个容器  
-d： 指定容器运行在后台  
--name：指定容器的名字  
--hostname：指定容器的主机名，和name不同，这个是其他容器加入其所在集群的标签。  
--net：指定Step 1创建的网络名  
-p 26257:26257 -p 8080:8080：指定内部节点和客户端节点的通信端口（26257）映射以及Admin界面从容器到主机的HTTP访问端口映射。  
-v "//c/Users/<username>/cockroach-data/roach1:/cockroach/cockroach-data"： 挂载一个主机目录作为容器的磁盘，意味着数据和日志将会存在Users/<username>/cockroach-data/roach1。更多细节可查看：[Docker's Bind Mounts](https://docs.docker.com/engine/admin/volumes/bind-mounts/)  
cockroachdb/cockroach:v2.0.5 start --insecure ：启动CockroachDB节点的命令

Step 3: 添加节点到集群

现在集群已经运行起来，但是集群只有一个节点，此时用户可以开始连接数据库并进行操作。在实际业务部署中，用户需要大于3个节点才能充分利用CockroachDB的自动冗余副本、数据均衡和集群容错的特性。

添加节点2

*#注意替换<username>成具体的内容*

PS C:\Users\username> docker run -d `

--name=roach2 `

--hostname=roach2 `

--net=roachnet `

-v "//c/Users/<username>/cockroach-data/roach2:/cockroach/cockroach-data" `

cockroachdb/cockroach:v2.0.5 start --insecure --join=roach1

添加节点3

*#注意替换<username>成具体的内容*

PS C:\Users\username> docker run -d `

--name=roach3 `

--hostname=roach3 `

--net=roachnet `

-v "//c/Users/<username>/cockroach-data/roach3:/cockroach/cockroach-data" `

cockroachdb/cockroach:v2.0.5 start --insecure --join=roach1

**TIPS:**  
-v：挂载一个主机目录作为容器的磁盘，数据和日志将会存在Users/<username>/cockroach-data/roach2或Users/<username>/cockroach-data/roach3  
--join：指定节点需要添加到集群的容器的hostname

Step 4: 测试集群

使用docker exec命令启动容器1对应的节点1的交互式shell：

PS C:\Users\username> docker exec -it roach1 ./cockroach sql --insecure

执行SQL操作：

**CREATE** **DATABASE** bank;

**CREATE** **TABLE** bank.accounts (id INT **PRIMARY** **KEY**, balance DECIMAL);

**INSERT** **INTO** bank.accounts **VALUES** (1, 1000.50);

**SELECT** \* **FROM** bank.accounts;

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

| id | balance |

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

| 1 | 1000.5 |

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

(1 row)

在容器2执行如下命令，启动节点2的交互式shell：

PS C:\Users\username> docker exec -it roach2 ./cockroach sql --insecure

查看数据：

**SELECT** \* **FROM** bank.accounts;

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

| id | balance |

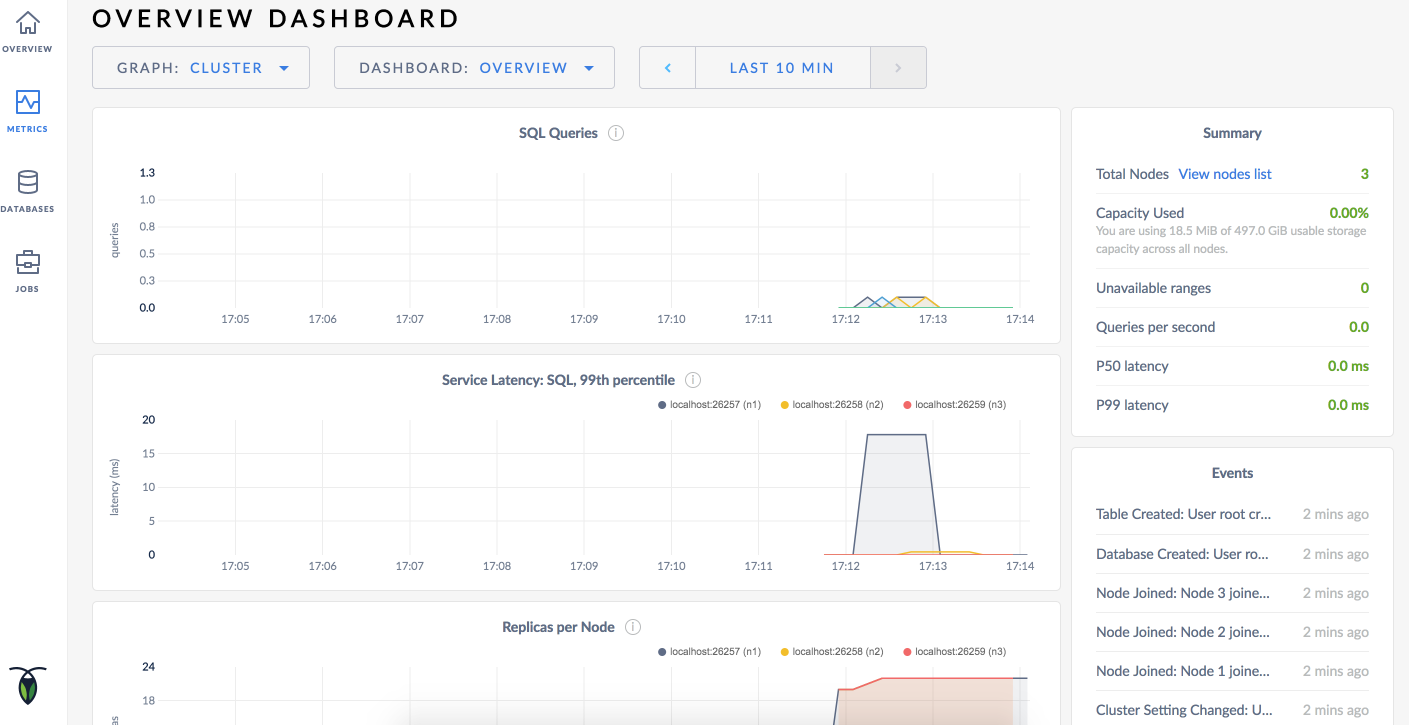
+*----+---------+*

| 1 | 1000.5 |

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

(1 row)

Step 5: 监控集群

先前已经将容器的8080端口和主机的8080端口做了映射，现在可以通过http://localhost:8080登陆Admin界面查看集群。 

默认情况下，CockroachDB对一份数据自动冗余3份副本到集群当中，均衡分布到各个节点，可在**Replicas per Node**页查看各节点数据冗余情况。

**TIPS:** 关于数据冗余、均衡分布、集群容错的更多介绍，可以查看：[数据复制](http://doc.cockroachchina.baidu.com/#quick-start/explore-benefits/data-replication)、[自动负载均衡](http://doc.cockroachchina.baidu.com/#quick-start/explore-benefits/automatic-rebalancing)、[容错及恢复](http://doc.cockroachchina.baidu.com/#quick-start/explore-benefits/fault-tolerance-and-recovery)

Step 6: 停止集群

使用docker stop和docker rm命令停止和删除容器

*# Stop the containers:*

PS C:\Users\username> docker stop roach1 roach2 roach3

*# Remove the containers:*

PS C:\Users\username> docker rm roach1 roach2 roach3

如不再需要集群数据，可以直接删除容器对应的主机目录

PS C:\Users\username> rm -rf cockroach-data