情境二：源站全掛掉，vs-b 自成 cluster

版本問題！

- 必須確保 vs-a & vs-b 的 redis 版本相同，不然會出現無法 migrate 的問題

基本邏輯

- vs-b 主機的三個 port 先脫離 slave 身分
- 修改 .conf 裡面的參數： `cluster-enabled` 改成 `yes`
- 逐一重啟 vs-b 主機的三個 port 服務。
- vs-b 主機的三個 port 組成一 cluster

```
## 注意：現在 UAT 的 lab 機器有兩個版本的 Redis
$ /usr/local/redis-cluster/bin/redis-server -v
Redis server v=7.0.5 sha=00000000:0 malloc=jemalloc-5.2.1 bits=64 build=1fa39f349e85dde3
$ /usr/local/redis/bin/redis-server -v
Redis server v=5.0.9 sha=00000000:0 malloc=jemalloc-5.1.0 bits=64 build=841c5ada03fd9517
```

```
## 先確認 b 站三服務的狀態

$ /usr/local/redis-cluster/bin/redis-cli -h `hostname` -p 6378 -c -a newld20200909 2> /dev/null info replication
# Replication
role:slave
master_host:10.21.1.120
master_port:6380
master_link_status:up
master_last_io_seconds_ago:9
master_sync_in_progress:0
slave_repl_offset:821785
slave_priority:100
slave_read_only:1
connected_slaves:1
slave0:ip=10.21.1.121,port=6380,state=online,offset=821785,lag=0
master_replid:15247563cec49ed3a91ea993b5edbf1011de1038
master_replid2:0000000000000000000000000000000000000000
master_repl_offset:821785
second_repl_offset:-1
repl_backlog_active:1
repl_backlog_size:1048576
repl_backlog_first_byte_offset:821702
repl_backlog_histlen:84


$ /usr/local/redis-cluster/bin/redis-cli -h `hostname` -p 6380 -c -a newld20200909 2> /dev/null info replication
# Replication
role:slave
master_host:10.21.1.121
master_port:6380
master_link_status:up
master_last_io_seconds_ago:4
master_sync_in_progress:0
slave_repl_offset:821841
slave_priority:100
slave_read_only:1
connected_slaves:0
master_replid:15247563cec49ed3a91ea993b5edbf1011de1038
master_replid2:0000000000000000000000000000000000000000
master_repl_offset:821841
second_repl_offset:-1
repl_backlog_active:1
repl_backlog_size:1048576
repl_backlog_first_byte_offset:821828
repl_backlog_histlen:14
```

```
$ /usr/local/redis-cluster/bin/redis-cli -h `hostname` -p 6382 -c -a newld20200909 2> /dev/null info replication
# Replication
role:slave
master_host:10.21.1.122
master_port:6380
master_link_status:up
master_last_io_seconds_ago:5
master_sync_in_progress:0
slave_repl_offset:821788
slave_priority:100
slave_read_only:1
connected_slaves:0
master_replid:ffe545e9486e086b6beed14ced37d9ce1abe1cca
master_replid2:0000000000000000000000000000000000000000
master_repl_offset:821788
second_repl_offset:-1
repl_backlog_active:1
repl_backlog_size:1048576
repl_backlog_first_byte_offset:821705
repl_backlog_histlen:84
```

```
## b 站三服務都先脫離 Slave 的狀態
$ /usr/local/redis-cluster/bin/redis-cli -h `hostname` -p 6378 -c -a newld20200909 2> /dev/null replicaof no one
OK
$ /usr/local/redis-cluster/bin/redis-cli -h `hostname` -p 6380 -c -a newld20200909 2> /dev/null replicaof no one
OK
$ /usr/local/redis-cluster/bin/redis-cli -h `hostname` -p 6382 -c -a newld20200909 2> /dev/null replicaof no one
OK

## 把 cluster-enabled no 修改為 cluster-enabled yes
$ vim /opt/centos/redis-6378.conf
$ vim /opt/centos/redis-6380.conf
$ vim /opt/centos/redis-6382.conf
```

```
## 重啟 b 站服務
## 關閉服務
netstat -nutlp | grep -e :63 | awk '{print $7}' | awk -F '/' '{print $1}' | xargs kill

## 移除各 nodes 底下的配置文件 nodes.conf & 資料文件 dump.rdb & aof
mv /usr/local/redis-cluster/data/dump.rdb /usr/local/redis-cluster/data/dump.rdb.origin
mv /usr/local/redis-cluster/data/appendonly.aof /usr/local/redis-cluster/data/appendonly.aof.origin
mv /usr/local/redis-cluster/data/nodes-6378.conf /usr/local/redis-cluster/data/nodes-6378.conf.origin
mv /usr/local/redis-cluster/data/nodes-6380.conf /usr/local/redis-cluster/data/nodes-6380.conf.origin
mv /usr/local/redis-cluster/data/nodes-6382.conf /usr/local/redis-cluster/data/nodes-6382.conf.origin
## 啟動服務：Redis 5.0.9 的版本
/usr/local/redis/bin/redis-server /opt/centos/redis-6378.conf
/usr/local/redis/bin/redis-server /opt/centos/redis-6380.conf
/usr/local/redis/bin/redis-server /opt/centos/redis-6382.conf
```

```
## 新建 cluster
$ /usr/local/redis-cluster/bin/redis-cli --cluster create 10.21.1.3:6378 10.21.1.3:6380 10.21.1.3:6382 --cluster-replicas 0 -a newld2020090
Warning: Using a password with '-a' or '-u' option on the command line interface may not be safe.
>>> Performing hash slots allocation on 3 nodes...
Master[0] -> Slots 0 - 5460
Master[1] -> Slots 5461 - 10922
Master[2] -> Slots 10923 - 16383
M: 09bce4a72a59d7ce9cf2dc9c22f0ef0ef8ae0c8e 10.21.1.3:6378
   slots:[0-5460] (5461 slots) master
M: ea29398c590af3c1a4e7a923120854dc5cc6f669 10.21.1.3:6380
   slots:[5461-10922] (5462 slots) master
M: 1abc780e4303c0e8f41d000df33b54b8c200874a 10.21.1.3:6382
   slots:[10923-16383] (5461 slots) master
Can I set the above configuration? (type 'yes' to accept): yes
>>> Nodes configuration updated
>>> Assign a different config epoch to each node
>>> Sending CLUSTER MEET messages to join the cluster
Waiting for the cluster to join
.
>>> Performing Cluster Check (using node 10.21.1.3:6378)
M: 09bce4a72a59d7ce9cf2dc9c22f0ef0ef8ae0c8e 10.21.1.3:6378
   slots:[0-5460] (5461 slots) master
M: 1abc780e4303c0e8f41d000df33b54b8c200874a 10.21.1.3:6382
   slots:[10923-16383] (5461 slots) master
```

```
M: ea29398c590af3c1a4e7a923120854dc5cc6f669 10.21.1.3:6380
   slots:[5461-10922] (5462 slots) master
[OK] All nodes agree about slots configuration.
>>> Check for open slots...
>>> Check slots coverage...
[OK] All 16384 slots covered.
```

```
## 把原本的 dump.rdb & appendonly.aof 放回
mv /usr/local/redis-cluster/data/dump.rdb.origin /usr/local/redis-cluster/data/dump.rdb
mv /usr/local/redis-cluster/data/appendonly.aof.origin /usr/local/redis-cluster/data/appendonly.aof

## 重啟 b 站服務
## 關閉服務
netstat -nutlp | grep -e :63 | awk '{print $7}' | awk -F '/' '{print $1}' | xargs kill
## 啟動服務：Redis 5.0.9 的版本
/usr/local/redis/bin/redis-server /opt/centos/redis-6378.conf
/usr/local/redis/bin/redis-server /opt/centos/redis-6380.conf
/usr/local/redis/bin/redis-server /opt/centos/redis-6382.conf
```

```
#### 先關閉源站 cluster 的所有服務
## 分別登入 10.21.1.120 & 10.21.1.121 & 10.21.1.122 執行

$ sudo su
$ sh /opt/centos/stop_redis_all.sh
停止服務中
已成功停止服務

## 確認相關 port 6377 & 6380 都已經關閉
$ netstat -ntulp | grep :63
tcp        0      0 0.0.0.0:6379            0.0.0.0:*               LISTEN      17753/./redis-serve
tcp6       0      0 :::6379                 :::*                    LISTEN      17753/./redis-serve

## vs-b 的三個 instance 要脫離 slave 身分
$ /usr/local/redis-cluster/bin/redis-cli -h `hostname` -p 6378 -c -a newld20200909 2> /dev/null replicaof no one
OK
$ /usr/local/redis-cluster/bin/redis-cli -h `hostname` -p 6380 -c -a newld20200909 2> /dev/null replicaof no one
OK
$ /usr/local/redis-cluster/bin/redis-cli -h `hostname` -p 6382 -c -a newld20200909 2> /dev/null replicaof no one
OK


## 進入 vs-b 各 instance，修改 conf 檔案
## 主要是把 cluster-enabled yes 變成不是註解
$ vim /opt/centos/redis-6378.conf
$ vim /opt/centos/redis-6380.conf
$ vim /opt/centos/redis-6382.conf

## 停止 vs-b 各 instance 的服務
$ netstat -nutlp | grep -e :63 | awk '{print $7}' | awk -F '/' '{print $1}' | xargs kill
$ netstat -nutlp | grep -e :63
tcp        0      0 0.0.0.0:6379            0.0.0.0:*               LISTEN      1429/./redis-server
tcp6       0      0 :::6379                 :::*                    LISTEN      1429/./redis-server

## 移除各 nodes 底下的配置文件 nodes.conf & 資料文件 dump.rdb
mv /usr/local/redis-cluster/data/dump.rdb /usr/local/redis-cluster/data/dump.rdb.origin
mv /usr/local/redis-cluster/data/nodes-6378.conf /usr/local/redis-cluster/data/nodes-6378.conf.origin
mv /usr/local/redis-cluster/data/nodes-6380.conf /usr/local/redis-cluster/data/nodes-6380.conf.origin
mv /usr/local/redis-cluster/data/nodes-6382.conf /usr/local/redis-cluster/data/nodes-6382.conf.origin

## 重啟 vs-b 各 instance 的服務
$ /usr/local/redis-cluster/bin/redis-server /opt/centos/redis-6378.conf
```

```
$ /usr/local/redis-cluster/bin/redis-server /opt/centos/redis-6380.conf
$ /usr/local/redis-cluster/bin/redis-server /opt/centos/redis-6382.conf
## 查看 vs-b 各 instance 的服務是否都順利啟動
$ netstat -nutlp | grep -e :63
tcp        0      0 10.21.1.3:6378          0.0.0.0:*               LISTEN      23303/redis-server
tcp        0      0 0.0.0.0:6379            0.0.0.0:*               LISTEN      1429/./redis-server
tcp        0      0 10.21.1.3:6380          0.0.0.0:*               LISTEN      23315/redis-server
tcp        0      0 10.21.1.3:6382          0.0.0.0:*               LISTEN      23327/redis-server
tcp6       0      0 :::6379                 :::*                    LISTEN      1429/./redis-server
```

```
### 針對 vs-b 的三個 instance 建立 cluster

$ /usr/local/redis-cluster/bin/redis-cli --cluster create 10.21.1.3:6378 10.21.1.3:6380 10.21.1.3:6382 --cluster-replicas 0 -a newld2020090



## 進入任一 port 查看 cluster 狀態
$ /usr/local/redis-cluster/bin/redis-cli -h `hostname` -p 6378 -c -a newld20200909 2> /dev/null cluster nodes
/usr/local/redis-cluster/bin/redis-cli -h `hostname` -p 6378 -c -a newld20200909 2> /dev/null cluster nodes
289cd799d35d8362a87c94057879bb111875e57a 10.21.1.3:6382@16382 slave fb7a8a516331499dc0cce6e754651192487a2692 0 1673584080000 15 connected
23a405fd0d07732e0117bb367f9e0b850ebfb14b 10.21.1.120:6377@16377 master,fail? - 1673584038779 1673584035853 16 disconnected 0-5461
6de8c57e166e6987e4f6473181d237c261ce41e5 10.21.1.3:6378@16378 myself,master - 0 1673584081000 0 connected [9940-<-b96b5dd558e7568f383aa4ab2
9cc6fcd015145cfe8bf81150bd58b4b8397e07e0 10.21.1.121:6380@16380 slave,fail? 23a405fd0d07732e0117bb367f9e0b850ebfb14b 1673584041800 16735840
959f6718cefd96214ec46025e835f9b405802983 10.21.1.3:6380@16380 slave b96b5dd558e7568f383aa4ab2fa625edac4bd020 0 1673584082088 17 connected
b96b5dd558e7568f383aa4ab2fa625edac4bd020 10.21.1.121:6377@16377 master,fail? - 1673584037772 1673584035853 17 disconnected 5462-10922
f7def362c61ae67a2025f8f820062c40d8061743 10.21.1.120:6380@16380 slave,fail? fb7a8a516331499dc0cce6e754651192487a2692 1673584043412 16735840
04349595dc7b856fd27a2c87d33c842cd506e1f2 10.21.1.122:6380@16380 slave,fail? b96b5dd558e7568f383aa4ab2fa625edac4bd020 1673584040793 16735840
fb7a8a516331499dc0cce6e754651192487a2692 10.21.1.122:6377@16377 master,fail? - 1673584042807 1673584035854 15 disconnected 10923-16383

## 脫離先前的 cluster
```

參考資料

redis主从同步，总是显示master_link_status:down的解决方法-阿里云开发者社区

查看了redis日志，发现日志里出现很多的"I/O error trying to sync with MASTER:connection lost"' 百度了下，发现是client-output-buffer-limit slave这个参数配置不当造成的。 网上找到的相关说明： # 这个参数太小的话，错误日志会显示I/O error trying to sync withMASTER: connection lost' # 这是说负责发数据给slave的client，如果buffer超过

https://developer.aliyun.com/article/503045