Linux MHA 세팅 및 사용법 A to Z

이 문서에서는 Virtual Box로 Linux 환경을 생성하고, MySQL을 설치하여 MHA 세팅을 진행하는 일련의 과정을 설명합니다. 문서의 구조는 아래와 같습니다.

--------------------------------------------------------------------------------------------------------------

1. 제목명 // 과정의 순번과 제목명  
   설명 // 해당 제목에 대한 설명  
   소스 코드(표) // 표 안에 관련 소스코드 내용이 정리되며 소스 코드 첫 부분에는 (Source), (Replica), (Manager), (mysql), # 등이 붙을 수 있습니다.

--------------------------------------------------------------------------------------------------------------

1. VirtualBox 설치  
   VirtualBox 다운로드 링크: www.virtualbox.org/   
   MHA 세팅을 위해서는 3대의 Linux가 필요한데, 그 환경을 Virtual Machine을 이용하여 만들어줍니다.
2. Ubuntu 20.04 설치  
   Ubuntu 20.04 다운로드 링크: https://ubuntu.com/download/desktop  
   2-1) VirtualBox 실행 -> 새로 만들기 -> Linux, Ubuntu 64bit로 생성  
   2-2) 설정 -> 저장소 -> 컨트롤러: IDE의 비어있음 부분 클릭 -> 광학 드라이브 옆 디스크(원형) 모양 클릭 -> 다운로드한 Ubuntu 20.04.iso (디스크 이미지) 파일 실행  
   2-3) 설정 -> 디스플레이 -> 그래픽 컨트롤러 VBoxVGA로 변경 후, 시작 버튼 클릭하여 Linux 환경 세팅 진행  
   ‘VBoxVGA’로 변경하는 이유는 처음 Linux 세팅할 때, 일부 버튼이 위치한 화면이 잘리면서 안보이는 현상이 발생하기 때문임. ‘VBoxVGA’로 변경하는 대신 ALT+F7로 이동하며 세팅도 가능.  
   2-4) 설정 -> 네트워크 -> ‘NAT’를 ‘어댑터에 브리지’로 변경  
   어댑터에 브리지로 변경해야 ip를 할당 받아 내부망을 사용할 수 있음
3. 초기 Linux 세팅 및 MySQL 설치  
   모든 설치가 진행되면 update, upgrade 등의 작업과 mysql 설치가 필요함

|  |
| --- |
| (Source, Replica, manager) su - // root 권한으로 접속  # apt update // #이 붙으면 root 권한을 의미 (vi 에디터의 #은 주석을 의미하니 주의)  # apt upgrade  # apt install mysql-server  # mysql\_secure\_installation // mysql\_secure\_installation은 선택 사항임  (disallow root login remotely 는 N, 나머지는 모두 Y로 설정)  # service mysql restart // 세팅 진행 후, 적용하기 위해서 service 명령으로 restart 해줘야 함.  # service mysql status // mysql 실행 상태 확인, # mysql로 실행도 가능  # apt net-tools // 추후 ifconfig, netstat 등의 기능 사용을 위해서 미리 설치 |

1. replication을 위한 mysql config 파일 설정  
   사용자마다 mysql의 config 파일 위치가 다를 수 있으나, /etc/mysql/mysql.conf.d/mysqld.cnf, /etc/mysql/my.cnf 등의 파일명을 가지고 있음.

|  |
| --- |
| (Source) # vi /etc/mysql/my.cnf // 여기선 vi 에디터로 my.cnf 파일 수정, nano 에디터도 무관.  [mysqld]  gtid\_mode=ON  enforce\_gtid\_consistency=ON  server\_id=1 // source server는 1, replica server는 2 (중복되면 에러 발생, MHA에선 11, 12 사용)  log\_bin=binlog // log파일명 (본인이 지정하고 싶은 파일명) |

|  |
| --- |
| (Replica) # vi /etc/mysql/my.cnf  [mysqld]  gtid\_mode=ON  enforce\_gtid\_consistency=ON  server\_id=2 // source server와는 절대 겹치면 안됨  relay\_log=testrelaylog // log 파일명 (본인이 지정하고 싶은 파일명)  relay\_log\_purge=ON  read\_only=ON // replication 상태에서는 write 기능을 사용하지 않고 read만 사용함  log\_slave\_updates=ON |

|  |
| --- |
| (Source) # vi /etc/mysql/mysql.conf.d/mysqld.cnf  # bind-address = 127.0.0.1 // 주석처리. 내부망에서만 사용하는 옵션이므로 주석처리함  # mysqlx-bind-address = 127.0.0.1 // 주석처리 |

|  |
| --- |
| (Replica) # vi /etc/mysql/mysql.conf.d/mysqld.cnf  # bind-address = 127.0.0.1 // replica 서버도 마찬가지로 주석처리해줌  # mysqlx-bind-address = 127.0.0.1 // 주석처리 |

1. replication을 위한 mysql 설정  
   새로운 user를 생성하고 replication 관련 권한을 부여함. 이 user를 통해 replica server에서 source server로 접근하여 replication 진행

|  |
| --- |
| (Source) mysql>  SET GLOBAL validate\_password.policy=LOW; // LOW는 비밀번호 길이 조건만 존재  CREATE USER ‘repluser’@’%’ IDENTIFIED BY ‘repl\_user\_password’;  GRANT REPLICATION SLAVE ON \*.\* TO ‘repl\_user’@’%’; |

|  |
| --- |
| (Replica) mysql>  CHANGE REPLICATION SOURCE TO  SOURCE\_HOST=’172.30.1.X’, // Source 서버의 ip 주소  SOURCE\_PORT=3306 // mysql의 기본 port는 3306  SOURCE\_USER=’repl\_user’, // source에서 생성한 유저 id  SOURCE\_PASSWORD=’repl\_user\_password’, // source에서 생성한 유저의 패스워드  SOURCE\_AUTO\_POSITION=1,  GET\_SOURCE\_PUBLIC\_KEY=1; |

1. mysql 설정 적용을 위한 재시작 및 replication 실행

|  |
| --- |
| (Source, Replica) # service mysql restart // 재시작 |

|  |
| --- |
| (Replica) # mysql  mysql> start replica;  mysql> show replica status \G; // replication 상태 확인 |

|  |
| --- |
| (Source) # mysql  mysql> show processlist \G; |

replica의 결과로 Error가 없으면 정상인 것이고, source의 결과로 replica에서 접근한 ip가 보이면 정상적으로 replication 되는 것입니다.

1. MHA Manager 및 MHA Node 다운로드  
   Manager Server에는 MHA Node와 MHA Manager를 설치해야 하며, Source, Replica Server에는 MHA Node만 설치가 필요합니다.   
   각 서버에서 아래 링크에 접속하여 Ubuntu에서 필요한 .deb 파일을 다운받아 준비합니다.  
   MHA Manager 다운로드 링크: https://github.com/yoshinorim/mha4mysql-manager/releases  
   MHA Node 다운로드 링크: https://github.com/yoshinorim/mha4mysql-node/releases/tag/v0.58

|  |
| --- |
| (Manager) su -  # apt-get install libdbd-mysql-perl // 패키지 의존성 문제가 있으므로, 가능하면  # apt-get install libconfig-tiny-perl // install 순서를 변경하지 않는 것을 추천함  # apt-get install liblog-dispatch-perl  # apt-get install libparallel-forkmanager-perl  # cd 다운로드 // 본인이 Manager, Node의 deb 파일 다운받은 경로로 이동  # dpkg -i mha4mysql-node\_X.Y\_all.deb // 본인이 다운한 버전에 맞춰 X, Y에 숫자 입력  # dpkg -i mha4mysql-manager\_X.Y\_all.deb |

|  |
| --- |
| (Source, Replica) su -  # apt-get install libdbd-mysql-perl  # cd 다운로드 // 본인이 Node deb 파일을 다운받은 경로  # dpkg -i mha4mysql-node\_X.Y.all.deb // 본인이 다운한 파일명에 맞춰서 X, Y 대체 |

1. Manager에서 Node 접근용 mysql 계정 생성  
   manager에서 source, replica server에 접근하기 위해서 source와 replica에 user를 생성해 줘야합니다. 현재 source와 replica는 replication 진행중이므로, source에서만 user를 생성해도, 자동으로 replica 쪽에 user가 생성이 됩니다.

|  |
| --- |
| (Source) mysql>  SET GLOBAL validate\_password.policy=LOW;  CREATE USER ‘mha’@’%’ identified by ‘mha\_password’;  GRANT ALL PRIVILEGES ON \*.\* TO ‘mha’@’%’;  FLUSH PRIVILEGES; |

1. MHA config 파일 설정

|  |
| --- |
| (manager) # vi /etc/app1.cnf  [server default]  user=mha // manager에서 source, replica DB에 접근할 user명  password=mhapassword  ssh\_user=root // ssh에서 접근할 때 사용할 user명(다음 단계에서 ssh 설정 진행함)  manager\_workdir=/var/log/masterha/app1  remote\_workdir=/var/log/masterha/app1  repl\_user=repl\_user // replication 설정한 user명  repl\_password=repl\_user\_password  [server1]  hostname=172.30.1.X // source server의 ip 주소  [server2]  hostname=172.30.1.Y // replica server의 ip 주소 |

1. SSH 접속 세팅  
   MHA를 사용하기 위해서는 manager server에서 SSH를 이용하여 source, replica server에 접근이 가능해야 합니다. SSH 접속을 위한 root 계정 비밀번호 변경, public key 생성 및 타 서버에 등록 등의 과정을 진행합니다.

|  |
| --- |
| (manager, source, replica) su -  # apt-get install ssh  # password root // root 계정 비밀번호 변경 (본인이 변경하고자하는 비밀번호 2회 입력)  0000, 0000  # ssh-keygen -t rsa -b 2048 // SSH key 생성 (rsa 방식의 2048 byte 길이 key 생성)  # vi /etc/ssh/sshd\_config // 아래 3개 항목 주석처리 제거 또는 직접 작성  PermitRootLogin yes // root로 접근 허용하는지  PubkeyAuthentication yes // ssh key 등록 방식의 인증 허용하는지  UseLogin yes  # ssh-copy-id root@172.30.1.XYZ // 이동하려는 PC의 IP 모두 한번씩 입력  manager에서는 source, replica의 ip 입력, source는 manager, replica / replica는 manager, source  # |

1. SSH 정상 유무 확인

|  |
| --- |
| (manager, source, replica) su -  # service ssh restart // 경우에 따라 sshd도 restart 필요할 수도 있음  (manager) // manager에서만 체크 가능  # masterha\_check\_ssh --conf=/etc/app1.cnf |

1. MHA 사용을 위한 설정 파일 수정  
   현재 우리가 설치한 MHA는 과거에 제작된 버전이기 때문에, Ubuntu 20.04, mysql 8.0.X 버전에서는 오류가 발생합니다. 오류를 제거하기 위해서 MHA 소스코드에 작성된 일부 코드들을 수정해줘야 합니다. (123)은 123번째 row를 의미하고, (+)는 추가, (-)는 제거를 의미합니다.

|  |
| --- |
| (manager) # vi /usr/share/perl5/MHA/DBHelper.pm // 경로가 다를 수 있으니, 오류 로그 확인!  (155) sub connect\_util { // 155번째줄 근처의 sub connect\_util 함수 수정  …  my $password = shift;  (-) my $dsn\_host = $host =~ m{:} ? ‘[‘ . $host . ‘]’ : $host; // 이것을 아래로 수정  (+) my $dsn\_host = $host =~ m{:} ? $host : $host; // [ ]가 포함되면 에러 발생하니 수정해줌  … }  (199) sub connect {  …  unless ( $self->{dsn} ) {  (-) my $dsn\_host = $host =~ m{:} ? ‘[‘ . $host . ‘]’ : $host;  (+) my $dsn\_host = $host =~ m{:} ? $host : $host;  .... } |

|  |
| --- |
| (manager) # vi /usr/share/perl5/MHA/HealthCheck.pm  (97) sub connect {  …  My $log = $self->{logger};  (-) My $dsn\_host = $self->{ip} =~ m{:} ? ‘[‘ . $self->{ip} . ‘]’ : $self->{ip};  (+) my $dsn\_host = $self->{ip} =~ m{:} ? $self->{ip} : $self->{ip};  … } |

|  |
| --- |
| (manager) # vi /usr/share/perl5/MHA/NodeUtil.pm  (194~) sub parse\_mysql\_version($) {  My $str =shift;  (+) $str =~ /(\d+)\.(\d+)/; // 추가  (+) my $strmajor = “$1.$2”; // 추가  (-) my $result = sprintf( ‘%03d%03d03d’, $str =~ m/(\d+)/g ); // 제거  (+) my $result = sprintf( ‘%03d%03d’, $strmajor =~ m/(\d+)/g ); // 추가  … }  (199~) sub parse\_mysql\_major\_version($) {  My $str = shift;  (+) $str =~ /(\d+)\.(\d+)/;  (+) my $strmajor = “$1.$2”;  (-) my $result = sprintf( ‘%03d%03d’, $str =~ m/(\d+)/g );  (+) my $result = sprintf( ‘%03d%03d’, $strmajor =~ m/(\d+)/g );  … } |

|  |
| --- |
| (manager) # vi /usr/share/perl5/MHA/SlaveUtil.pm  (77~) sub get\_relay\_log\_info\_type {  …  (-) my $type;  $mysql\_version = get\_version($dbh) unless ($mysql\_version);  (-) if ( !is\_mariadb($mysql\_version) && MHA::NodeUtil::mysql\_version\_ge( $mysql\_version, “5.6.2” ) ) {  (-) $type = get\_variable( $dbh, Get\_Relay\_Log\_Info\_Type\_SQL ); }  (-) unless ( defined($type) ) {  (-) $type = “FILE”; }  (+) if ( !MHA::NodeUtil::mysql\_version\_ge( $mysql\_version, “8.0.0” ) ) {  (+) return “FILE”; }  (+) my $type = get\_variable( $dbh, Get\_Relay\_Log\_Info\_Type\_SQL );  Return $type; } |

1. MHA replication 기능 확인  
   종종 이렇게 세팅했는데도 replication 에러가 발생할 수 있습니다. 그것을 해결하기 위해서, 미리 manager에서 source와 replica server의 mysql에 접속을 한번 해줍니다.

|  |
| --- |
| (manager) su -  # mysql -umha -pmhapassword -h 172.30.1.X -P 3306 // -h에는 source의 IP를 적어주면 되고, -P에는 mysql의 포트를 입력해줍니다.  mysql > exit  # mysql -umha -pamhapassword -h 172.30.1.Y -P 3306 // 이번에는 replica의 mysql에 접속해줍니다.  mysql > exit  # masterha\_check\_repl --conf=/etc/app1.cnf |

1. MHA 실행  
   MHA replicaion까지 기능이 OK이면, manager를 실행할 수 있습니다. 실행했을 때, command 창이 막히면서 “ ~ and sleeping until it doesn’t respond” 와 같은 응답이 나타납니다. 에러가 아니니 놀라지 않아도 괜찮습니다.

|  |
| --- |
| (manager) # masterha\_manager --conf=/etc/app1.cnf |

1. MHA 정상 확인 및 정지  
   그래도 manager가 정상인지 확인하고 싶다면 masterha\_check\_status를 통해 정상 유무를 판단할 수 있습니다.

|  |
| --- |
| (manager) # masterha\_check\_status --conf=/etc/app1.cnf // 새로운 터미널을 실행해 입력합니다.  (manager) # masterha\_stop --conf=/etc/app1.cnf // MHA의 실행을 중지시킵니다. |

1. MHA VIP 세팅 진행  
   VIP는 가상 IP를 의미하며, Spring 소스코드 변경을 최소화 하기 위해 사용합니다. Spring 소스 코드에 VIP로 접근하여 CRUD 동작한다고 설정해 놓으면, Source server의 DB에 문제가 생기더라도 Replica Server에 VIP를 할당하면 되므로, Spring 소스 코드는 수정할 필요가 없습니다.

|  |
| --- |
| (master) # ifconfig enp0s3:0 172.30.1.Z up // enp0s3:0에 172.30.1.Z라는 가상 IP 설정  (master) # ifconfig enp0s3:0 down // enp0s3:0의 가상 IP 설정 해제 |

1. MHA script 수정 및 작성  
   일반적으로 master\_ip\_failover와 master\_ip\_online\_change 스크립트가 사용됩니다. 이 문서에서는 두 스크립트를 수정하기는 하나, 실질적으로는 master\_ip\_failover만 사용됩니다.  
   다운 링크: https://github.com/yoshinorim/mha4mysql-manager/tree/master/samples/scripts  
   위의 다운로드 링크에서 두 파일을 다운받아서 적절한 경로로 이동시켜줍니다.  
   경로는 9에서 작성한 /etc/app1.cnf에서 작성된 workdir과 동일합니다.  
   스크립트를 작성하고 권한이 없으면 권한을 꼭 설정해줘야 합니다. (ls -al, chmod 명령 사용)  
   MHA script는 failover 동작 시 수행될 스크립트를 작성하는 것이며, VIP 변경 등이 가능합니다.

|  |
| --- |
| (manager) su -  # cd /var/log/masterha / scripts 폴더 생성을 위해 해당 경로로 이동  # mkdir scripts / scripts 폴더가 없을 경우에만 생성  # mv /기존파일위치 /var/log/masterha/scripts // master\_ip\_failover 파일을 scripts 폴더로 이동 |

|  |
| --- |
| (manager) # vi /var/log/masterha/scripts/master\_ip\_failover  (96) ## Creating an app user on the new master // 96라인 근처의 아래 소스코드 주석처리  ## print "Creating app user on the new master..n"; // 주석처리 진행  ## FIXME\_xxx\_create\_user( $new\_master\_handler->{dbh} );  ## $new\_master\_handler->enable\_log\_bin\_local();  ## $new\_master\_handler->disconnect();  ## Update master ip on the catalog database, etc  ## FIXME\_xxx;  system(“/bin/bash /var/log/masterha/scripts/mha\_change\_vip.sh”); // 소스코드 추가 |

|  |
| --- |
| (manager) # vi /var/log/masterha/scripts/master\_ip\_online\_change  (149라인, 244라인 주석처리)  ## Drop application user so that nobody can connect. Disabling per-session binlog beforehand  ## $orig\_master\_handler->disable\_log\_bin\_local();  ## print current\_time\_us() . " Drpping app user on the orig master..n";  ## FIXME\_xxx\_drop\_app\_user($orig\_master\_handler); // 149라인 주석처리  ## Creating an app user on the new master // 244라인 주석처리  ## print current\_time\_us() . " Creating app user on the new master..n";  ## FIXME\_xxx\_create\_app\_user($new\_master\_handler);  ## $new\_master\_handler->enable\_log\_bin\_local();  ## $new\_master\_handler->disconnect();  ## Update master ip on the catalog database, etc  ## vip change // 각 상황에 맞게 vip 변경 여기서는 28이 replica, 51이 source 서버임  If ( $new\_master\_ip eq “172.30.1.28” ) { // 사용자에 맞게 28을 본인 replica ip로 변경  System(“/bin/bash /var/log/masterha/scripts/change\_vip\_slave\_to\_master.sh”);  }  Elsif ( $new\_master\_ip eq “172.30.1.51” ) { // 사용자에 맞게 51을 본인 replica ip로 변경  System(“/bin/bash /var/log/masterha/scripts/change\_vip\_master\_to\_slave.sh”);  }  $exit\_code = 0; … |

|  |
| --- |
| (manager) # vi /var/log/masterha/scripts/mha\_change\_vip.sh  #!/bin/bash  ssh root@172.30.1.X ifconfig enp0s3:0 down // X에는 source의 ip  ssh root@172.30.1.Y ifconfig enp0s3:0 172.30.1.Z up // Y는 replica의 ip, Z는 VIP의 ip설정값 |

|  |
| --- |
| (manager) # vi /var/log/masterha/scripts/change\_vip\_master\_to\_slave.sh  #!/bin/bash  ssh root@172.30.1.X ifconfig enps3:0 down // X에는 source의 ip  ssh root@172.30.1.Y ifconfig enps3:0 172.30.1.Z up // Y에는 replica의 ip, Z는 VIP의 ip설정값 |

|  |
| --- |
| (manager) # vi /var/log/masterha/scripts/change\_vip\_slavae\_to\_master.sh  #!/bin/bash  ssh root@172.30.1.Y ifconfig enp0s3:0 down // Y는 replica의 ip  ssh root@172.30.1.X ifconfig enp0s3:0 172.30.1.Z up // X는 source의 ip, Z는 VIP의 ip 설정값 |

|  |
| --- |
| (manager) su -  # cd /var/log/masterha/scripts  # ls - al // ls - al의 결과로 스크립트 파일들은 권한이 744로 설정되어 있어야함  # chmod 744 master\_ip\_failover\_script // 권한이 7이 아니면, bash 파일 실행이 안됨  # chmod 744 master\_ip\_online\_change\_script |

1. MHA config 파일 수정   
   위에서 작성한 script를 사용하기 위해서, MHA config 파일에 script 경로를 설정해줘야 합니다.

|  |
| --- |
| (manager) # vi /etc/app1.cnf  [server default]  User = ~..  …  (+) master\_ip\_failover\_script=/var/log/masterha/scripts/master\_ip\_failover // 추가  (+) master\_ip\_online\_change\_script=/var/log/masterha/scripts/master\_ip\_online\_change // 추가  [server1]  … |

1. failover 발생  
   failover되는 조건은 어떠한 문제로 인해서 Source 서버의 mysql에 접속이 안되는 경우입니다. Source 서버의 mysql을 종료시키면 manager 서버에서 동작중인 MHA가 Source 서버의 이상을 확인하고, failover를 수행합니다.

|  |
| --- |
| (manager) # masterha\_manager --conf=/etc/app1.cnf // MHA를 실행합니다. |

|  |
| --- |
| (Source) # service mysql stop // Source 서버의 mysql을 종료시킵니다. |

1. failover 동작 확인  
   failover가 제대로 이루어지면, Source와 Replica 서버 간의 replication이 끊어집니다. 또한, Replica의 ReadOnly 설정이 풀리면서 Write 명령에도 에러가 발생하지 않습니다.

|  |
| --- |
| (Source) # service mysql start  # mysql  mysql > show processlist \G; // replica 서버 정보가 존재하지 않아야 정상입니다. |

|  |
| --- |
| (Replica) # mysql  mysql > show replica status \G; // empty set이 결과로 도출되어야 정상입니다. |

1. failover 원상복구 1단계 (데이터 동기화)  
   failover가 진행되면, Source 서버와 Replica 서버의 replication 연결이 끊어지고 Replica 서버가 Master 권한을 가져 데이터가 쌓이게 됩니다.  
   다시 원상복구 하기 위해서는 Replica 서버의 데이터를 Source 서버로 이동시키고, 다시 Replication을 걸어줘야 합니다.  
   원상 복구는 Replication을 반대로(기존: Replica에서 Source를 설정, 원복: Source에서 Replica를 설정) 걸어줘서 진행하게 됩니다. 그 과정에서 만약에 오류가 발생하면 아래와 같이 Source 서버에서 접근 가능한 user를 Replica 서버의 mysql에 생성해줍니다.

|  |
| --- |
| (Replica) mysql> // 에러 발생하는 경우에만 생성하면 되고, 아니면 생략  SET GLOBAL validate\_password.policy=LOW; // LOW는 비밀번호 길이 조건만 존재  CREATE USER ‘repluser’@’172.30.1.X’ IDENTIFIED BY ‘repl\_user\_password’;  GRANT REPLICATION SLAVE ON \*.\* TO ‘repl\_user’@’172.30.1.X’; |

|  |
| --- |
| (Source) # mysql  CHANGE REPLICATION SOURCE TO  SOURCE\_HOST=’172.30.1.Y’, // Replica 서버의 ip 주소  SOURCE\_PORT=3306, // 기본 PORT 는 3306  SOURCE\_USER=’repl\_user’,  SOURCE\_PASSWORD=’repl\_user\_password’,  SOURCE\_AUTO\_POSITION=1,  GET\_SOURCE\_PUBLIC\_KEY=1; |

1. failover 원상복구 2단계 (MHA 설정 원상복구)  
   failover로 인해 VIP, replication 등의 설정이 변경되었으므로, 수정해줘야 합니다.  
   1) 데이터 동기화 과정에서 반대로 설정된 Source와 Replica의 권한 변경이 필요합니다.   
   2) 변경된 VIP 설정을 원상 복구해줘야 합니다. (VIP를 Source에 할당되게 변경)  
   3) Source와 Replica의 replicaion을 재설정 해줘야합니다.

|  |
| --- |
| (manager) # > rm /var/log/masterha/app1/app1.failover/complete // failover 기록 삭제  # masterha\_master\_switch --master\_state=alive --conf=/etc/app1.cnf // 권한 변경 |

|  |
| --- |
| (manager) # bash /var/log/masterha/scripts/change\_vip\_slave\_to\_master.sh // VIP 변경 |

|  |
| --- |
| (Source) # mysql  mysql > CHANGE REPLICATION SOUCRE TO  SOURCE\_HOST=’172.30.1.X’, // Source 서버의 ip  SOURCE\_PORT=3306, // 기본 포트는 3307  SOURCE\_USER=’repl\_user’,  SOURCE\_PASSWORD=’repl\_user\_password’,  SOURCE\_AUTO\_POSITION=1,  GET\_SOURCE\_PUBLIC\_KEY=1;  mysql > start replica; // replication 시작  mysql > show replica status \G; // replication 상태 확인 |

1. MHA 재실행  
   다시 원상복구가 정상적으로 이루어졌으므로, MHA를 다시 실행시켜줍니다. 실행 방법은 기존과 동일합니다.

|  |
| --- |
| (manager) # masterha\_manager --conf=/etc/app1.cnf |

* 기타 참고 사이트 리스트

master\_check\_repl 에러 관련: https://github.com/yoshinorim/mha4mysql-manager/issues/89  
제작자 Github: https://github.com/yoshinorim/mha4mysql-manager/wiki  
SSH 에러 해결: https://konkuk-curry.tistory.com/entry/Error-permission-denied-publickey-password-lost-connection  
MHA 세팅: https://linux.systemv.pe.kr/m-ha-mysql-설정/  
master\_ip\_failover 허가 오류: https://adbancedteam.tistory.com/192  
MHA 세팅(MariaDB + CentOS + MHA0.57): https://bstar36.tistory.com/367  
MHA 세팅: https://khj93.tistory.com/entry/MHA-MHA-구성-및-설치-DB이중화-Fail-Over-테스트  
DB replication 관련 정리: https://server-talk.tistory.com/241

※ [리눅스 패키지 의존성] dpkg: error processing archive 에러 해결 방법

# apt-get install -f // 에러 확인 목적으로 실행

※ masterha\_check\_repl --conf=/etc/app1.cnf 했을 때

[error] [/ usr / local / share / perl / 5.18.2 / MHA / MasterMonitor.pm, ln427] Error happened on checking configurations. at / usr / local / share / perl /5.18.2/MHA/MasterMonitor.pm line 329.

라는 에러가 뜨면 권한 설정이 제대로 안 된 것이므로 소스-레플리케이션 권한 설정을 다시 한다.

※ masterha\_check\_ssh --conf=/etc/app1.cnf 했을 때 에러가 나면

mha - source - replica 사이에 비밀번호 없이 ssh 접속이 모두 되는지 확인한다.