[HCI 이중화 구성 (drbd+PostgreSQL+pacemaker)](https://wiki.gsretail.com/pages/viewpage.action?pageId=90853318)

[Skip to end of metadata](https://wiki.gsretail.com/pages/viewpage.action?pageId=90853318#page-metadata-end)

* 작성자 : [신영진/클라우드팀](https://wiki.gsretail.com/display/~sahra), 최근 변경 : [백승엽/클라우드팀](https://wiki.gsretail.com/display/~sybaek) - [2022-04-26](https://wiki.gsretail.com/pages/diffpagesbyversion.action?pageId=90853318&selectedPageVersions=11&selectedPageVersions=12)

[Go to start of metadata](https://wiki.gsretail.com/pages/viewpage.action?pageId=90853318#page-metadata-start)

**01. HCI 초기 구성**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **1) 초기 파티션(Master / Standby 동일 구성)**  **DRBD 설치**   |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10 | # lsblk  NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT  sda 8:0 0 120G 0 disk  ├─sda1 8:1 0 1G 0 part /boot  └─sda2 8:2 0 116G 0 part  ├─centos-LV\_ROOT 253:0 0 50G 0 lvm /  ├─centos-LV\_SWAP 253:1 0 16G 0 lvm [SWAP]  └─centos-LV\_APPLICATIONS 253:2 0 50G 0 lvm /applications  sdb 8:16 0 50G 0 disk  sr0 11:0 1 1024M 0 rom |   **2) 파티션 설정(DRBD로 사용할 볼륨)**  **파티션 설정**   |  |  | | --- | --- | | 1  2  3  4 | # fdisk /dev/sdb (파티션 전체 할당 : n -> p -> 1 -> Enter -> Enter -> w)  # pvcreate /dev/sdb1  # vgcreate vg01 /dev/sdb1  # lvcreate -l 100%FREE -n LV\_DATA vg01 |   **3) 최종 파티션 (Master / Standby 동일 구성)**  **DRBD 설정\_리소스 설정**   |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | # lvs  LV VG Attr LSize Pool Origin Data% Meta% Move Log Cpy%Sync Convert  LV\_APPLICATIONS centos -wi-ao---- 50.00g  LV\_ROOT centos -wi-ao---- 50.00g  LV\_SWAP centos -wi-ao---- <16.00g  LV\_DATA vg01 -wi-a----- <50.00g    # lsblk  NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT  sda 8:0 0 120G 0 disk  ├─sda1 8:1 0 1G 0 part /boot  └─sda2 8:2 0 116G 0 part  ├─centos-LV\_ROOT 253:0 0 50G 0 lvm /  ├─centos-LV\_SWAP 253:1 0 16G 0 lvm [SWAP]  └─centos-LV\_APPLICATIONS 253:2 0 50G 0 lvm /applications  sdb 8:16 0 50G 0 disk  └─sdb1 8:17 0 50G 0 part  └─vg01-LV\_DATA 253:3 0 50G 0 lvm  sr0 11:0 1 1024M 0 rom | |

**02. DRBD설치**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1) DRBD 설치**  **DRBD 설치**   |  |  | | --- | --- | | 1  2  3 | # rpm --import <https://www.elrepo.org/RPM-GPG-KEY-elrepo.org>  # rpm -Uvh <http://elrepo.org/elrepo-release-7.0-3.el7.elrepo.noarch.rpm>  # yum install kernel-devel drbd84-utils kmod-drbd84 |   **2) Config & Resource 설정**  **DRBD 설정\_리소스 설정**   |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57 | # cd /etc/drbd.d/  # cp -a global\_common.conf global\_common.conf.bak <-default 파일 백업    # vi global\_common.conf    # DRBD is the result of over a decade of development by LINBIT.  # In case you need professional services for DRBD or have  # feature requests visit [http://www.linbit.com](http://www.linbit.com/)    global {            usage-count yes;            # minor-count dialog-refresh disable-ip-verification  }  common {            handlers {                    pri-on-incon-degr       "/usr/lib/drbd/notify-pri-on-incon-degr.sh; /usr/lib/drbd/notify-emergency-reboot.sh; echo b > /proc/sysrq-trigger ; reboot -f";                    pri-lost-after-sb       "/usr/lib/drbd/notify-pri-lost-after-sb.sh; /usr/lib/drbd/notify-emergency-reboot.sh; echo b > /proc/sysrq-trigger ; reboot -f";                    fence-peer      /usr/lib/drbd/crm-fence-peer.sh;                    local-io-error  "/usr/lib/drbd/notify-io-error.sh; /usr/lib/drbd/notify-emergency-shutdown.sh; echo o > /proc/sysrq-trigger ; halt -f";                    split-brain     "/usr/lib/drbd/notify-split-brain.sh root";                    after-resync-target     /usr/lib/drbd/crm-unfence-peer.sh;            }            startup {                    degr-wfc-timeout        120;                    outdated-wfc-timeout    120;            }            net {                    max-epoch-size  8000;                    max-buffers     8000;                    unplug-watermark    16;                    sndbuf-size     512K;                    after-sb-0pri   discard-zero-changes;                    after-sb-1pri   discard-secondary;            }            disk {                    on-io-error     detach;                    fencing resource-only;                    al-extents      3389;                    resync-rate 300M;            }  }    # vi dbcluster.res    resource dbcluster {       volume 0 {               device          /dev/drbd0;               disk            /dev/mapper/vg01-LV\_DATA;               flexible-meta-disk      internal;            }       on hsnetcondb146p {          address      10.52.232.174:7788;       }       on hsnetcondb147p {          address      10.52.232.175:7788;       }  } |   **3) DRBD initialization & Meta data 생성 (All Node)**  **DRBD 설정\_메타데이터생성**   |  |  | | --- | --- | | 1 | # drbdadm create-md dbcluster |   **참고사항**   - Metadata 생성 전 DRBD볼륨으로 사용할 공간은 파일시스템 생성 전이어야 함  -  Command 'drbdmeta 1 v08 /dev/xxx internalcreate-md' terminated with exit code 40 메시지 발생 할 경우      dd if=/dev/zero of=/dev/sdb bs=1M count=128  **4) Load the DRBD kernel module (All Node)**  **DRBD 설정\_DRBD모듈 인식**   |  |  | | --- | --- | | 1  2  3  4  5 | # modprobe drbd    # lsmod | grep drbd    drbd                  317306  4    libcrc32c               1246  1 drbd |   **5) DRBD(dbcluster) attach and connect (All Node)**  **DRBD attach and connect (All Node)**   |  |  | | --- | --- | | 1  2 | # drbdadm up dbcluster (all node)  # drbdadm -- --overwrite-data-of-peer primary dbcluster (node 01) |   **6) DRBD(dbcluster) attach and connect check(All Node)**  **DRBD attach and connect (All Node)**   |  |  | | --- | --- | | 1  2  3  4  5  6  7  8 | # cat /proc/drbd    version: 8.4.11-1 (api:1/proto:86-101)  GIT-hash: 66145a308421e9c124ec391a7848ac20203bb03c build by mockbuild@, 2020-04-05 02:58:18   0: cs:SyncSource ro:Primary/Secondary ds:UpToDate/Inconsistent C r-----      ns:816128 nr:0 dw:0 dr:818248 al:8 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:51606940          [>....................] sync'ed:  1.6% (50396/51192)M          finish: 0:08:25 speed: 102,016 (102,016) K/sec |   **7) Change drbd node**  **DRBD 상태 확인**   |  |  | | --- | --- | | 1  2 | # drbdadm primary dbcluster  # drbdadm secondary dbcluster |   **8) Filesystem Format & Mount (Primary 서버)**  **DRBD filesystem 포맷 & 마운트(primary)**   |  |  | | --- | --- | | 1  2  3 | # mkfs -t xfs /dev/drbd0 (node01)  # mkdir -p /pgdb/data  # mount -t xfs /dev/drbd0 /pgdb/data (node01) | |

**02. PostgreSQL 설치(DBA)**

|  |
| --- |
| **1)**[**PostgreSQL Binary 설치**](https://wiki.gsretail.com/pages/viewpage.action?pageId=65712471) |

**03. PCS 설치 및 설정 (Pacemaker)**

|  |
| --- |
|  |
| **1) PCS 설치**  **PCS 설치**   |  |  | | --- | --- | | 1  2  3  4  5  6  7  8 | 1. pcs 설치  yum install pcs    2. 이중화 계정 패스워드 설정  passwd hacluster    3. 데몬 기동  systemctl enable pcsd --now |   **2) hosts파일 설정**  **hosts파일 설정**   |  |  | | --- | --- | | 1  2  3  4 | # ev list  10.52.233.61    gfevdbvip061p  10.52.233.62    gfevdbm1-062p  10.52.233.63    gfevdbm2-063p |   **3) PCS 설정**  **PCS 설정**   |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45 | pcs cluster auth gfevdbm1-062p gfevdbm2-063p  Username: hacluster     (PWD 동일)  Password:  gfevdbm1-062p: Authorized  gfevdbm2-063p: Authorized    pcs cluster setup --name gfevdb-cluster gfevdbm1-062p gfevdbm2-063p    pcs cluster start --all    pcs cluster cib clust\_cfg    pcs -f clust\_cfg property set stonith-enabled=false    pcs -f clust\_cfg property set no-quorum-policy=ignore    pcs -f clust\_cfg resource defaults resource-stickiness=1000    pcs -f clust\_cfg resource create gfevdb ocf:linbit:drbd drbd\_resource=dbcluster op monitor interval=30s    pcs -f clust\_cfg resource master gfevdbClone gfevdb master-max=1 master-node-max=1 clone-max=2 clone-node-max=1 notify=true    pcs -f clust\_cfg resource create filesystem Filesystem device="/dev/drbd0" directory="/pgdb/data" fstype="xfs"    pcs -f clust\_cfg constraint colocation add filesystem with gfevdbClone INFINITY with-rsc-role=Master    pcs -f clust\_cfg constraint order promote gfevdbClone then start filesystem    pcs -f clust\_cfg resource create vip ocf:heartbeat:IPaddr2 ip=10.52.233.61 cidr\_netmask=24 op monitor interval=30s    pcs -f clust\_cfg resource create pgsql pgsql pgctl="/applications/pgdb/app/pgsql/11.15/bin/pg\_ctl" psql="/applications/pgdb/app/pgsql/11.15/bin/psql" pgdata="/pgdb/data" config="/pgdb/data/postgresql.conf" restart\_on\_promote='true' op start  timeout="60s" interval="0s"  on-fail="restart" op monitor timeout="60s" interval="4s" on-fail="restart" op monitor timeout="60s" interval="3s"  on-fail="restart" role="Master" op promote timeout="60s" interval="0s"  on-fail="restart" op demote  timeout="60s" interval="0s"  on-fail="stop" op stop    timeout="60s" interval="0s"  on-fail="block"    pcs -f clust\_cfg constraint colocation add pgsql with filesystem INFINITY     ## pgsql 과 filesystem 리소스는 같은 노드에서 실행 설정    pcs -f clust\_cfg constraint order filesystem then pgsql      ## filesystem 리소스 실행 후 pgsql 리소스 실행 순서 설정    pcs -f clust\_cfg constraint colocation add vip with pgsql INFINITY     ## vip 리소스와 pgsql 리소스가 같은 노드에서 실행 설정    pcs -f clust\_cfg constraint order vip then  pgsql     ## vip 리소스 시작 후 pgsql 리소스 실행 순서 설정    pcs -f clust\_cfg constraint    pcs -f clust\_cfg resource show    pcs cluster cib-push clust\_cfg |   **4) PCS 상태확인**  **PCS 상태확인**   |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25 | [root@gfevdbm1-062p drbd.d]# pcs status  Cluster name: gfevdb-cluster  Stack: corosync  Current DC: gfevdbm1-062p (version 1.1.23-1.el7\_9.1-9acf116022) - partition with quorum  Last updated: Tue Apr 26 10:39:18 2022  Last change: Tue Apr 26 10:33:30 2022 by root via cibadmin on gfevdbm2-063p    2 nodes configured  5 resource instances configured    Online: [ gfevdbm1-062p gfevdbm2-063p ]    Full list of resources:     Master/Slave Set: gfevdbClone [gfevdb]       Masters: [ gfevdbm1-062p ]       Slaves: [ gfevdbm2-063p ]   filesystem     (ocf::heartbeat:Filesystem):    Started gfevdbm1-062p   vip    (ocf::heartbeat:IPaddr2):       Started gfevdbm1-062p   pgsql  (ocf::heartbeat:pgsql): Started gfevdbm1-062p    Daemon Status:    corosync: active/disabled    pacemaker: active/disabled    pcsd: active/enabled |   **참고사항**   - failover 명령어     pcs node standby node명     pcs node unstandby node명(해제)  **참고사항**   - Maintenance Mode(Freez) 상태  [참고] 유지보수 상태로 DB작업 시 Cluster 상관 없이 작업 가능 함(Fail-over 안됨)   * Maintenance Mode(Freez) 상태 설정   # pcs property set maintenance-mode   -> "Resource management is DISABLED" 메시지 및 Resources의 (unmanaged) 상태로 전환 확인   |  | | --- | | # crm\_mon  Stack: corosync Current DC: hstcomdb144p-hb (version 1.1.19-8.el7-c3c624ea3d) - partition with quorum Last updated: Mon Dec 24 14:13:21 2018 Last change: Mon Dec 24 14:12:56 2018 by root via cibadmin on hstcomdb144p-hb  2 nodes configured 10 resources configured                \*\*\* Resource management is DISABLED \*\*\*   The cluster will not attempt to start, stop or recover services  Online: [ hstcomdb144p-hb hstcomdb145p-hb ]  Active resources:  hstcomdb144p-fence       (stonith:fence\_ipmilan):            Started hstcomdb144p-hb (unmanaged) hstcomdb145p-fence       (stonith:fence\_ipmilan):            Started hstcomdb144p-hb (unmanaged)   Resource Group: hstcomdb-service      hstcomdb-vip       (ocf::heartbeat:IPaddr2):           Started hstcomdb144p-hb (unmanaged)      hstcomdb-oravg     (ocf::heartbeat:LVM):               Started hstcomdb144p-hb (unmanaged)      hstcomdb-data       (ocf::heartbeat:Filesystem):        Started hstcomdb144p-hb (unmanaged)      hstcomdb-arch       (ocf::heartbeat:Filesystem):        Started hstcomdb144p-hb (unmanaged)      hstcomdb-oracle    (ocf::heartbeat:oracle):            Started hstcomdb144p-hb (unmanaged)      hstcomdb-listener  (ocf::heartbeat:oralsnr):           Started hstcomdb144p-hb (unmanaged)      Clone Set: bond0-monitor-clone [bond0-monitor] (unmanaged)      bond0-monitor       (ocf::heartbeat:ethmonitor):        Started hstcomdb145p-hb (unmanaged)      bond0-monitor       (ocf::heartbeat:ethmonitor):        Started hstcomdb144p-hb (unmanaged) |  * Maintenance Mode(Freez) 상태 해제   # pcs property unset maintenance-mode |

[Skip to end of metadata](https://wiki.gsretail.com/pages/viewpage.action?pageId=65712515#page-metadata-end)

* [이준성/클라우드팀](https://wiki.gsretail.com/display/~leejs1229)님이 작성, [2022-03-17](https://wiki.gsretail.com/pages/diffpagesbyversion.action?pageId=65712515&selectedPageVersions=3&selectedPageVersions=4)에 최종 변경

[Go to start of metadata](https://wiki.gsretail.com/pages/viewpage.action?pageId=65712515#page-metadata-start)

PostgreSQL 글로벌 개발 그룹은 일년에 한번 정도 주요 버전을 출시

주요 버전은 새로운 기능 포함 및 버그 수정이 포함되며 필요한 경우 마이너 릴리즈 (3개월에 한번) 진행

* 마이너 릴리즈
  + 매 분기마다 최소 한 번의 마이너 릴리즈 목표 (최소 4번 정도)
  + 2, 5, 8, 11월의 두 번째 목요일 릴리즈
  + 기본 마이너 릴리즈는 위와 같으며 중요 이슈 발생 시 비 정기적으로 릴리즈

중요 버그 또는 보안 수정이 필요한 경우 예정 일정 이외 릴리즈 진행

주요 버전은 초기 릴리즈 후 5년 동안 지원

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Current minor** | **Supported** | **First Release** | **Final Release** |
| **14** | **14.2** | **Yes** | **2021년 09월 30일** | **2026년 11월 12일** |
| **13** | **13.6** | **Yes** | **2020년 09월 24일** | **2025년 11월 13일** |
| **12** | **12.1** | **Yes** | **2019년 10월 03일** | **2024년 11월 14일** |
| **11** | **11.15** | **Yes** | **2018년 10월 18일** | **2023년 11월 09일** |
| 10 | 10.2 | Yes | 2017년 10월 05일 | 2022년 11월 10일 |
| 9.6 | 9.6.24 | No | 2016년 09월 29일 | 2021년 11월 11일 |
| 9.5 | 9.5.25 | No | 2016년 01월 07일 | 2021년 02월 11일 |
| 9.4 | 9.4.26 | No | 2014년 12월 18일 | 2020년 02월 13일 |
| 9.3 | 9.3.25 | No | 2013년 09월 09일 | 2018년 11월 08일 |
| 9.2 | 9.2.24 | No | 2012년 09월 10일 | 2017년 11월 09일 |
| 9.1 | 9.1.24 | No | 2011년 09월 12일 | 2016년 10월 27일 |
| 9 | 9.0.23 | No | 2010년 09월 20일 | 2015년 10월 08일 |
| 8.4 | 8.4.22 | No | 2009년 07월 01일 | 2014년 07월 24일 |
| 8.3 | 8.3.23 | No | 2008년 02월 04일 | 2013년 02월 07일 |
| 8.2 | 8.2.23 | No | 2006년 12월 05일 | 2011년 12월 05일 |
| 8.1 | 8.1.23 | No | 2005년 11월 08일 | 2010년 11월 08일 |
| 8 | 8.0.26 | No | 2005년 01월 19일 | 2010년 10월 01일 |
| 7.4 | 7.4.30 | No | 2003년 11월 17일 | 2010년 10월 01일 |
| 7.3 | 7.3.21 | No | 2002년 11월 27일 | 2007년 11월 27일 |
| 7.2 | 7.2.8 | No | 2002년 02월 04일 | 2007년 02월 04일 |
| 7.1 | 7.1.3 | No | 2001년 04월 13일 | 2006년 04월 13일 |
| 7 | 7.0.3 | No | 2000년 05월 08일 | 2005년 05월 08일 |
| 6.5 | 6.5.3 | No | 1999년 06월 09일 | 2004년 06월 09일 |
| 6.4 | 6.4.2 | No | 1998년 10월 30일 | 2003년 10월 30일 |
| 6.3 | 6.3.2 | No | 1998년 03월 01일 | 2003년 03월 01일 |

**License**

| **PostgreSQL is released under the PostgreSQL License, a liberal Open Source license, similar to the BSD or MIT licenses.**  **PostgreSQL Database Management System (formerly known as Postgres, then as Postgres95)**  **Portions Copyright © 1996-2022, The PostgreSQL Global Development Group**  **Portions Copyright © 1994, The Regents of the University of California**  **Permission to use, copy, modify, and distribute this software and its documentation for any purpose, without fee, and without a written agreement is hereby granted, provided that the above copyright notice and this paragraph and the following two paragraphs appear in all copies.**  **IN NO EVENT SHALL THE UNIVERSITY OF CALIFORNIA BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS, ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN IF THE UNIVERSITY OF CALIFORNIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.**  **THE UNIVERSITY OF CALIFORNIA SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE SOFTWARE PROVIDED HEREUNDER IS ON AN "AS IS" BASIS, AND THE UNIVERSITY OF CALIFORNIA HAS NO OBLIGATIONS TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.** |
| --- |

# [복제 구성](https://wiki.gsretail.com/pages/viewpage.action?pageId=90863719)

[Skip to end of metadata](https://wiki.gsretail.com/pages/viewpage.action?pageId=90863719#page-metadata-end)

* [이준성/클라우드팀](https://wiki.gsretail.com/display/~leejs1229)님이 작성, [2022-09-29](https://wiki.gsretail.com/pages/diffpagesbyversion.action?pageId=90863719&selectedPageVersions=8&selectedPageVersions=9)에 최종 변경

[Go to start of metadata](https://wiki.gsretail.com/pages/viewpage.action?pageId=90863719#page-metadata-start)

**Streaming Replication**

 - 블록 주소와 바이트 단위를 사용하는 물리적 복제

 - 전체 클러스터 복제

 - 동일 OS 및 DB 버전 사용

**Logical Replication**

 - 복제 ID (일반적으로 Primary Key)를 기반으로 변경 사항을 복제

 - 게시 (Publish) 와 구독 (Subscribers) 모델 기반

 - 테이블 단위 복제 및 다른 버전의 OS 및 DB에서도 적용 가능

[Skip to end of metadata](https://wiki.gsretail.com/display/DB/Logical+Replication#page-metadata-end)

* [이준성/클라우드팀](https://wiki.gsretail.com/display/~leejs1229)님이 작성, [2022-09-29](https://wiki.gsretail.com/pages/diffpagesbyversion.action?pageId=127482495&selectedPageVersions=14&selectedPageVersions=15)에 최종 변경

[Go to start of metadata](https://wiki.gsretail.com/display/DB/Logical+Replication#page-metadata-start)

**환경 구성** 원본 접기

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45 | ## 사전 환경 설정    0. 복제 하려는 테이블에 PK 값이 있어야 가능 (없어도 가능 하다는데.. 힘들게 가지 맙시다)  0. 복제하려는 테이블 구조가 sub 에 생성 되어 있어야 함.    1. Replication 용 유저 생성 (대상 : Pub)   -  CREATE ROLE repl\_user WITH REPLICATION PASSWORD 'qwe123' LOGIN;    2. postgresql.conf 파일 수정 (대상 : Pub)   - wal\_level : logical   - max\_wal\_senders   - max\_replication\_slots   - wal\_keep\_segments    3. pg\_hba.conf 파일 수정 (대상 : Pub)    4. Rplication Slot 생성 (대상 : Pub, 옵션)   - CREATE SUBSCRIPTION 명령어 실행 시 자동으로 Pub 서버에 동일 이름의 slot 이 생성 되며 특정 slot 으로 하고 싶을 경우 아래 방법으로 생성      - 생성 : SELECT \* FROM pg\_create\_physical\_replication\_slot('repl\_slot\_01');      - 확인 : SELECT \* FROM pg\_replication\_slots;      ## Pub 서버 설정    1. 복제를 원하는 테이블에 대해 pub 설정   - CREATE PUBLICATION [PUB명] FOR TABLE [테이블명]...   - CREATE PUBLICATION pub\_tbs FOR TABLE gsefadm.tb\_md\_item\_ctns , gsefadm.tb\_md\_item\_prc;  ※ 확인 : pg\_publication, pg\_publication\_tables    샘플   - 모든 테이블 복제 : CREATE PUBLICATION alltables FOR ALL TABLES;   - INSERT 구문 만 복제 : CREATE PUBLICATION insert\_only FOR TABLE mydata WITH (publish = 'insert');    ## Sub 서버 설정  1. 구독 설정    - CREATE SUBSCRIPTION [SUB명] CONNECTION [PUB 연결 정보] PUBLICATION [PUB명];    - CREATE SUBSCRIPTION sub\_tbs CONNECTION 'host=10.52.233.202 user=logical\_repl passowrd=qwe123 port=5432 dbname=dsm' PUBLICATION pub\_tbs;  ※ 확인 : pg\_subscription ,  pg\_stat\_subscription  ※ 기타    - sub 설정 후 pub 서버에 sub명과 동일한 복제 슬롯 자동 생성 (지정해서 sub 구성 가능)    - 대용량 테이블의 경우 초기 동기화 시 네트워크 사용률 높아 질 수 있으며 동기화 완료 까지 소요 시간 발생      : 초기 동기화를 원하지 않을 경우 옵션 적용 가능      : CREATE SUBSCRIPTION sub\_tbs CONNECTION 'host=10.52.233.202 user=logical\_repl passowrd=qwe123 port=5432 dbname=dsm' PUBLICATION pub\_tbs WITH (copy\_data=FALSE);      : 이 경우 초기 동기 화 없이 복제 구성 완료 후 진행 되는 트랜잭션에 대해서만 복제    - sub 환경에서 데이터 수정 가능 (단, 데이터 어그러지면 자칫 더 복잡해짐) |

**유지 관리** 원본 접기

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28 | ## 복제 대상 추가/삭제  1. Pub 에 대상 테이블 추가   - ALTER PUBLICATION pub\_tbs ADD[DROP] TABLE gsefadm.tb\_md\_item\_dtl\_desc;  2. Sub 에서 refresh 실행 (미 실행 시 동기화 안됨)   - ALTER SUBSCRIPTION sub\_tbs REFRESH PUBLICSTION;      ## sub 대상 정보 확인  select  cla.relname,  sub.subname ,  sub.subenabled ,  sub.subconninfo ,  sub.subslotname ,  sub.subsynccommit ,  sub.subpublications ,  case when rel.srsubstate='i' then 'initialize'  when rel.srsubstate='d' then 'data is being copied'  when rel.srsubstate='f' then 'finished table copy'  when rel.srsubstate='s' then 'synchronized'  else 'ready'  end as state,  rel.srsublsn  from pg\_catalog.pg\_subscription sub,  pg\_catalog.pg\_subscription\_rel rel,  pg\_catalog.pg\_class cla  where sub.oid = rel.srsubid  and rel.srrelid = cla.oid; |

[Skip to end of metadata](https://wiki.gsretail.com/display/DB/Streaming+Replication#page-metadata-end)

* [이준성/클라우드팀](https://wiki.gsretail.com/display/~leejs1229)님이 작성, [2022-09-29](https://wiki.gsretail.com/pages/diffpagesbyversion.action?pageId=127482491&selectedPageVersions=1&selectedPageVersions=2)에 최종 변경

[Go to start of metadata](https://wiki.gsretail.com/display/DB/Streaming+Replication#page-metadata-start)

**환경 구성** 원본 접기

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44 | 1. Replication 용 유저 생성 (대상 : Master)   -  CREATE ROLE repl\_user WITH REPLICATION PASSWORD 'qwe123' LOGIN;    2. postgresql.conf 파일 수정 (대상 : Master)   - wal\_level : replica   - max\_wal\_senders   - max\_replication\_slots   - wal\_keep\_segments    3. pg\_hba.conf 파일 수정 (대상 : Master)    4. Rplication Slot 생성 (대상 : Master)      - 생성 : SELECT \* FROM pg\_create\_physical\_replication\_slot('repl\_slot\_01');      - 확인 : SELECT \* FROM pg\_replication\_slots;    5. Standby 서버 $PGDATA 경로 정리    6. pg\_basebackup 실행 (대상 : Standby 서버)  [postgres@gf-prd-dp-dev data]$ pg\_basebackup --host=10.52.221.94 --pgdata=/pgdb/data --username=repl\_user --write-recovery-conf --slot=repl\_slot\_01 --progress --verbose --max-rate=70M     ## pg\_basebackup 실행  pg\_basebackup: initiating base backup, waiting for checkpoint to complete  pg\_basebackup: checkpoint completed  pg\_basebackup: write-ahead log start point: E/28 on timeline 1  pg\_basebackup: starting background WAL receiver  38478911/38478911 kB (100%), 1/1 tablespace  pg\_basebackup: write-ahead log end point: E/4000050  pg\_basebackup: waiting for background process to finish streaming ...  pg\_basebackup: base backup completed  [postgres@gf-prd-dp-dev data]$    7. postgresql.conf 파일 수정 (대상 : Standby)   - hot\_standby = on   - host\_standby\_feedback = on    8. recovery.conf 파일 수정 (대상 : Standby)   - standby\_mode = on   - primary\_conninfo   ## pg\_basebackup 시 --write-recovery-conf 옵션 추가 시 자동 생성    9. Standby DB 서비스 실행   - Standby 서버 환경에 맞게 postgresql.conf 파일 수정 및 서비스 시작    00. 상태 확인   - Master : select \* from pg\_stat\_replication;   - Standby : select \* from pg\_stat\_wal\_receiver; |

**복제 연결이 깨질 경우** 원본 접기

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133 | Standby 서버 서비스 중지 시 Master 서버는 WAL 파일을 재 사용 하지 않고 보관  Master 서버의 WAL 파일 삭제 전 Standby 서버 복제가 다시 연결 되면 순차적으로 동기화 진행 및 WAL 파일 재사용    [postgres@gf-prd-evt-dev pg\_wal]$ ls -alh  합계 2.9G  drwx------  3 postgres postgres 4.0K  4월 19 09:34 .  drwx------ 20 postgres postgres 4.0K  4월 19 09:34 ..  -rw-------  1 postgres postgres  340  4월 18 18:56 000000010000000E0000002A.00000028.backup  -rw-------  1 postgres postgres  64M  4월 19 09:26 000000010000000F0000000B  -rw-------  1 postgres postgres  64M  4월 19 09:26 000000010000000F0000000C  -rw-------  1 postgres postgres  64M  4월 19 09:26 000000010000000F0000000D  -rw-------  1 postgres postgres  64M  4월 19 09:26 000000010000000F0000000E  -rw-------  1 postgres postgres  64M  4월 19 09:27 000000010000000F0000000F  -rw-------  1 postgres postgres  64M  4월 19 09:27 000000010000000F00000010  -rw-------  1 postgres postgres  64M  4월 19 09:27 000000010000000F00000011  -rw-------  1 postgres postgres  64M  4월 19 09:27 000000010000000F00000012  -rw-------  1 postgres postgres  64M  4월 19 09:30 000000010000000F00000013  -rw-------  1 postgres postgres  64M  4월 19 09:30 000000010000000F00000014  -rw-------  1 postgres postgres  64M  4월 19 09:30 000000010000000F00000015  -rw-------  1 postgres postgres  64M  4월 19 09:30 000000010000000F00000016  -rw-------  1 postgres postgres  64M  4월 19 09:30 000000010000000F00000017  -rw-------  1 postgres postgres  64M  4월 19 09:30 000000010000000F00000018  -rw-------  1 postgres postgres  64M  4월 19 09:30 000000010000000F00000019  -rw-------  1 postgres postgres  64M  4월 19 09:30 000000010000000F0000001A  -rw-------  1 postgres postgres  64M  4월 19 09:30 000000010000000F0000001B  -rw-------  1 postgres postgres  64M  4월 19 09:30 000000010000000F0000001C  -rw-------  1 postgres postgres  64M  4월 19 09:30 000000010000000F0000001D  -rw-------  1 postgres postgres  64M  4월 19 09:30 000000010000000F0000001E  -rw-------  1 postgres postgres  64M  4월 19 09:30 000000010000000F0000001F  -rw-------  1 postgres postgres  64M  4월 19 09:30 000000010000000F00000020  -rw-------  1 postgres postgres  64M  4월 19 09:30 000000010000000F00000021  -rw-------  1 postgres postgres  64M  4월 19 09:30 000000010000000F00000022  -rw-------  1 postgres postgres  64M  4월 19 09:30 000000010000000F00000023  -rw-------  1 postgres postgres  64M  4월 19 09:30 000000010000000F00000024  -rw-------  1 postgres postgres  64M  4월 19 09:30 000000010000000F00000025  -rw-------  1 postgres postgres  64M  4월 19 09:30 000000010000000F00000026  -rw-------  1 postgres postgres  64M  4월 19 09:30 000000010000000F00000027  -rw-------  1 postgres postgres  64M  4월 19 09:31 000000010000000F00000028  -rw-------  1 postgres postgres  64M  4월 19 09:31 000000010000000F00000029  -rw-------  1 postgres postgres  64M  4월 19 09:31 000000010000000F0000002A  -rw-------  1 postgres postgres  64M  4월 19 09:31 000000010000000F0000002B  -rw-------  1 postgres postgres  64M  4월 19 09:31 000000010000000F0000002C  -rw-------  1 postgres postgres  64M  4월 19 09:31 000000010000000F0000002D  -rw-------  1 postgres postgres  64M  4월 19 09:31 000000010000000F0000002E  -rw-------  1 postgres postgres  64M  4월 19 09:31 000000010000000F0000002F  -rw-------  1 postgres postgres  64M  4월 19 09:31 000000010000000F00000030  -rw-------  1 postgres postgres  64M  4월 19 09:31 000000010000000F00000031  -rw-------  1 postgres postgres  64M  4월 19 09:31 000000010000000F00000032  -rw-------  1 postgres postgres  64M  4월 19 09:31 000000010000000F00000033  -rw-------  1 postgres postgres  64M  4월 19 09:31 000000010000000F00000034  -rw-------  1 postgres postgres  64M  4월 19 09:32 000000010000000F00000035  -rw-------  1 postgres postgres  64M  4월 19 09:34 000000010000000F00000036  -rw-------  1 postgres postgres  64M  4월 19 09:34 000000010000000F00000037  -rw-------  1 postgres postgres  64M  4월 19 09:35 000000010000000F00000038  drwx------  2 postgres postgres 4.0K  4월 19 09:34 archive\_status      기타 서버 문제로 복제 연결이 끊어 지고 Master 서버 WAL 파일이 삭제 된 경우  Standby 서버에 DB로그에 아래와 같이 에러 로그 발생    [18279]: LOG:  entering standby mode  [18279]: LOG:  redo starts at F/C073DA8  [18279]: LOG:  consistent recovery state reached at F/2E683FA0  [18277]: LOG:  database system is ready to accept read only connections  [18294]: LOG:  started streaming WAL from primary at F/2C000000 on timeline 1  [18294]: FATAL:  could not receive data from WAL stream: ERROR:  requested WAL segment 000000010000000F0000000B has already been removed     ## Master 서버 pg\_wal 경로에 해당 wal 파일이 이미 삭제  [18295]: LOG:  started streaming WAL from primary at F/2C000000 on timeline 1  [18295]: FATAL:  could not receive data from WAL stream: ERROR:  requested WAL segment 000000010000000F0000000B has already been removed  [18310]: LOG:  started streaming WAL from primary at F/2C000000 on timeline 1  [18310]: FATAL:  could not receive data from WAL stream: ERROR:  requested WAL segment 000000010000000F0000000B has already been removed  아카이브 로그에서 pg\_wal 경로로 WAL 파일 복사하여 순차적으로 동기화 진행      ## Master 서버 : 위 에러로그에 해당되는 wal 파일을 찾아 pg\_wal에 복사 (아카이브 및 백업에서 wal 파일을 찾지 못할 경우 처음부터 재구성...)  [postgres@gf-prd-evt-dev archive]$ cp 000000010000000F00000010 /pgdb/data/pg\_wal/  [postgres@gf-prd-evt-dev archive]$ cp 000000010000000F00000011 /pgdb/data/pg\_wal/  [postgres@gf-prd-evt-dev archive]$ cp 000000010000000F00000012 /pgdb/data/pg\_wal/  [postgres@gf-prd-evt-dev archive]$ cp 000000010000000F00000013 /pgdb/data/pg\_wal/  [postgres@gf-prd-evt-dev archive]$ cp 000000010000000F00000014 /pgdb/data/pg\_wal/  [postgres@gf-prd-evt-dev archive]$ cp 000000010000000F00000015 /pgdb/data/pg\_wal/  [postgres@gf-prd-evt-dev archive]$ cp 000000010000000F00000016 /pgdb/data/pg\_wal/  [postgres@gf-prd-evt-dev archive]$ cp 000000010000000F00000017 /pgdb/data/pg\_wal/    ## Standby 서버 DB 로그 [18543]: LOG:  started streaming WAL from primary at F/58000000 on timeline 1  [18543]: FATAL:  could not receive data from WAL stream: ERROR:  requested WAL segment 000000010000000F00000019 has already been removed  [18547]: LOG:  started streaming WAL from primary at F/64000000 on timeline 1  [18547]: FATAL:  could not receive data from WAL stream: ERROR:  requested WAL segment 000000010000000F00000019 has already been removed  [18551]: LOG:  started streaming WAL from primary at F/64000000 on timeline 1  [18551]: FATAL:  could not receive data from WAL stream: ERROR:  requested WAL segment 000000010000000F0000001A has already been removed     ## 동기화 진행 하면서 순차적으로 다음 WAL 파일 요구  [18554]: LOG:  started streaming WAL from primary at F/68000000 on timeline 1  [18554]: FATAL:  could not receive data from WAL stream: ERROR:  requested WAL segment 000000010000000F0000001A has already been removed  [18558]: LOG:  started streaming WAL from primary at F/68000000 on timeline 1  [18558]: FATAL:  could not receive data from WAL stream: ERROR:  requested WAL segment 000000010000000F0000001A has already been removed  [18563]: LOG:  started streaming WAL from primary at F/68000000 on timeline 1  [18563]: FATAL:  could not receive data from WAL stream: ERROR:  requested WAL segment 000000010000000F0000001D has already been removed  [18568]: LOG:  started streaming WAL from primary at F/74000000 on timeline 1  [18568]: FATAL:  could not receive data from WAL stream: ERROR:  requested WAL segment 000000010000000F0000001D has already been removed  [18572]: LOG:  started streaming WAL from primary at F/74000000 on timeline 1  [18572]: FATAL:  could not receive data from WAL stream: ERROR:  requested WAL segment 000000010000000F0000001F has already been removed  [18575]: LOG:  started streaming WAL from primary at F/7C000000 on timeline 1  [18575]: FATAL:  could not receive data from WAL stream: ERROR:  requested WAL segment 000000010000000F0000001F has already been removed  [18579]: LOG:  started streaming WAL from primary at F/7C000000 on timeline 1  [18579]: FATAL:  could not receive data from WAL stream: ERROR:  requested WAL segment 000000010000000F00000020 has already been removed  [18582]: LOG:  started streaming WAL from primary at F/80000000 on timeline 1  [18582]: FATAL:  could not receive data from WAL stream: ERROR:  requested WAL segment 000000010000000F00000020 has already been removed  [18586]: LOG:  started streaming WAL from primary at F/80000000 on timeline 1  [18586]: FATAL:  could not receive data from WAL stream: ERROR:  requested WAL segment 000000010000000F00000020 has already been removed  [18591]: LOG:  started streaming WAL from primary at F/80000000 on timeline 1  [18591]: FATAL:  could not receive data from WAL stream: ERROR:  requested WAL segment 000000010000000F00000020 has already been removed  [18596]: LOG:  started streaming WAL from primary at F/80000000 on timeline 1  [18596]: FATAL:  could not receive data from WAL stream: ERROR:  requested WAL segment 000000010000000F00000020 has already been removed  [18601]: LOG:  started streaming WAL from primary at F/80000000 on timeline 1  [18601]: FATAL:  could not receive data from WAL stream: ERROR:  requested WAL segment 000000010000000F00000022 has already been removed  [18604]: LOG:  started streaming WAL from primary at F/88000000 on timeline 1  [18604]: FATAL:  could not receive data from WAL stream: ERROR:  requested WAL segment 000000010000000F00000023 has already been removed  [18608]: LOG:  started streaming WAL from primary at F/8C000000 on timeline 1      동기화 완료 시 상태 확인 가능 및 Master 서버 pg\_wal 경로의 wal 파일 재 사용되며 정리  [postgres@gf-prd-evt-dev pg\_wal]$ ls -rlth  합계 641M  -rw------- 1 postgres postgres 340  4월 18 18:56 000000010000000E0000002A.00000028.backup  -rw------- 1 postgres postgres 64M  4월 19 09:40 000000010000001000000005  -rw------- 1 postgres postgres 64M  4월 19 09:40 000000010000001000000006  -rw------- 1 postgres postgres 64M  4월 19 09:41 000000010000001000000007  -rw------- 1 postgres postgres 64M  4월 19 09:43 000000010000000F0000003E  -rw------- 1 postgres postgres 64M  4월 19 09:43 000000010000000F0000003F  -rw------- 1 postgres postgres 64M  4월 19 09:43 000000010000001000000000  -rw------- 1 postgres postgres 64M  4월 19 09:43 000000010000001000000001  -rw------- 1 postgres postgres 64M  4월 19 09:43 000000010000001000000002  -rw------- 1 postgres postgres 64M  4월 19 09:43 000000010000001000000003  drwx------ 2 postgres postgres 281  4월 19 09:43 archive\_status  -rw------- 1 postgres postgres 64M  4월 19 09:43 000000010000001000000004  [postgres@gf-prd-evt-dev pg\_wal]$ |

# [설치](https://wiki.gsretail.com/pages/viewpage.action?pageId=65712471)

[Skip to end of metadata](https://wiki.gsretail.com/pages/viewpage.action?pageId=65712471#page-metadata-end)

* 작성자 : [이준성/클라우드팀](https://wiki.gsretail.com/display/~leejs1229), 최근 변경 : [문광모/클라우드팀](https://wiki.gsretail.com/display/~kwangmo.moon) - [2022-09-28](https://wiki.gsretail.com/pages/diffpagesbyversion.action?pageId=65712471&selectedPageVersions=11&selectedPageVersions=12)

[Go to start of metadata](https://wiki.gsretail.com/pages/viewpage.action?pageId=65712471#page-metadata-start)

0. PostgreSQL 14.2 버전 기준

1. PostgreSQL RPM 패키지 설치

* 다운로드 : <https://yum.postgresql.org/14/redhat/rhel-7-x86_64/repoview/postgresqldbserver14.group.html> 해당 사이트에서 모든 RPM 패키지(4EA) 다운로드
* 설치 : 개별 설치 시 아래 순으로 설치
  + postgresql14-libs - The shared libraries required for any PostgreSQL clients
  + postgresql14 - PostgreSQL client programs and libraries
  + postgresql14-server - The programs needed to create and run a PostgreSQL server
  + postgresql14-contrib - Contributed source and binaries distributed with PostgreSQL

2. 소스 파일 설치

* 다운로드 : <https://www.postgresql.org/ftp/source/v14.2/>

**OS 계정 생성 및 프로파일 설정** 원본 접기

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32 | dbtest03:(MySQLTest3):/home]sudo -i  dbtest03:():/root]useradd -m -U postgres  dbtest03:():/root]cat /etc/passwd     ~ 생략 ~    mysql:x:510:510::/home/mysql:/bin/bash  00512038:x:1005:1005::/home/00512038:/bin/bash  pmm-agent:x:509:509:pmm-agent:/usr/local/percona/:/bin/false  postgres:x:1006:1006::/home/postgres:/bin/bash  dbtest03:():/root]passwd postgres  postgres 사용자의 비밀 번호 변경 중  새 암호:  잘못된 암호: 암호에는 1개 미만의 대문자가 포함되어 있습니다  새 암호 재입력:  passwd: 모든 인증 토큰이 성공적으로 업데이트 되었습니다.  dbtest03:():/root]        ## OS 계정(postgres) 프로파일 설정    ## PostgreSQL 14.2  export PGHOME=/applications/pgdb/app/pgsql/14.2  export PGDATA=/applications/pgdb/data  export LD\_LIBRARY\_PATH=/applications/pgdb/app/lib  export MANPATH=/pgdb/app/share/man      PATH=$PATH:$HOME/.local/bin:$HOME/bin:$PGHOME/bin    export PATH |

**프로그램 및 데이터파일 경로 설정** 원본 접기

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25 | dbtest03:():/applications]mkdir pgdb  dbtest03:():/applications]ls  pgdb/  dbtest03:():/applications]ls -alh  합계 0  drwxr-xr-x 3 service wheel 18 3월 18 15:03 ./  dr-xr-xr-x. 21 root root 287 2월 8 16:57 ../  drwxr-xr-x 2 root root 6 3월 18 15:03 pgdb/  dbtest03:():/applications]chown -R postgres:postgres ./pgdb/  dbtest03:():/applications]ls -alh  합계 0  drwxr-xr-x 3 service wheel 18 3월 18 15:03 ./  dr-xr-xr-x. 21 root root 287 2월 8 16:57 ../  drwxr-xr-x 2 postgres postgres 6 3월 18 15:03 pgdb/  dbtest03:():/applications]    [postgres@dbtest03 pgdb]$ mkdir app  [postgres@dbtest03 pgdb]$ mkdir data  [postgres@dbtest03 pgdb]$ ls -alh  합계 0  drwxr-xr-x 4 postgres postgres 29 3월 18 15:04 .  drwxr-xr-x 3 service wheel 18 3월 18 15:03 ..  drwxr-xr-x 2 postgres postgres 6 3월 18 15:04 app  drwxr-xr-x 2 postgres postgres 6 3월 18 15:04 data  [postgres@dbtest03 pgdb]$ |

**프로그램 컴파일 및 설치** 원본 접기

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138 | [postgres@dbtest03 InstallFile]$ tar -xvf postgresql-14.2.tar.gz     ## 압축 소스 파일 압축 해제  postgresql-14.2/  postgresql-14.2/.dir-locals.el  postgresql-14.2/contrib/  postgresql-14.2/contrib/tcn/  postgresql-14.2/contrib/tcn/tcn.control  postgresql-14.2/contrib/tcn/Makefile  postgresql-14.2/contrib/tcn/tcn.c     ~ 생략 ~    postgresql-14.2/Makefile  postgresql-14.2/README  postgresql-14.2/COPYRIGHT  postgresql-14.2/GNUmakefile.in  postgresql-14.2/.gitattributes  postgresql-14.2/aclocal.m4  postgresql-14.2/INSTALL  [postgres@dbtest03 InstallFile]$        [postgres@dbtest03 postgresql-14.2]$ ./configure --prefix=/applications/pgdb/app/pgsql/14.2     ## 설치 경로 지정  checking build system type... x86\_64-pc-linux-gnu  checking host system type... x86\_64-pc-linux-gnu  checking which template to use... linux  checking whether NLS is wanted... no  checking for default port number... 5432  checking for block size... 8kB  checking for segment size... 1GB  checking for WAL block size... 8kB  checking for gcc... no  checking for cc... no  configure: error: in `/home/postgres/InstallFile/postgresql-14.2':  configure: error: no acceptable C compiler found in $PATH     ## yum install gcc 설치  See `config.log' for more details  [postgres@dbtest03 postgresql-14.2]$ TMOUT=0  [postgres@dbtest03 postgresql-14.2]$        [postgres@dbtest03 postgresql-14.2]$ ./configure --prefix=/applications/pgdb/app/pgsql/14.2  checking build system type... x86\_64-pc-linux-gnu  checking host system type... x86\_64-pc-linux-gnu  checking which template to use... linux  checking whether NLS is wanted... no  checking for default port number... 5432    ~ 생략 ~    checking for library containing backtrace\_symbols... none required  checking for library containing gethostbyname\_r... none required  checking for library containing pthread\_barrier\_wait... -lpthread  checking for library containing readline... no  configure: error: readline library not found     ## yum install readline-devel 설치  If you have readline already installed, see config.log for details on the  failure. It is possible the compiler isn't looking in the proper directory.  Use --without-readline to disable readline support.  [postgres@dbtest03 postgresql-14.2]$        checking for library containing readline... -lreadline  checking for inflate in -lz... no  configure: error: zlib library not found     ## yum install zlib-devel 설치  If you have zlib already installed, see config.log for details on the  failure. It is possible the compiler isn't looking in the proper directory.  Use --without-zlib to disable zlib support.  [postgres@dbtest03 postgresql-14.2]$        [postgres@dbtest03 postgresql-14.2]$make world     ## world 구문까지 입력해야 관련 문서 등 부가적인 부분도 포함    ~ 생략 ~    make[2]: Leaving directory `/home/postgres/InstallFile/postgresql-14.2/contrib/vacuumlo'  make[1]: Leaving directory `/home/postgres/InstallFile/postgresql-14.2/contrib'  [postgres@dbtest03 postgresql-14.2]$        [postgres@dbtest03 postgresql-14.2]$ make install-world     ## 프로그램 설치  make -C ./src/backend generated-headers  make[1]: Entering directory `/home/postgres/InstallFile/postgresql-14.2/src/backend'  make -C catalog distprep generated-header-symlinks    ~ 생략 ~    /usr/bin/install -c -m 644 ./unaccent.rules '/applications/pgdb/app/pgsql/14.2/share/tsearch\_data/'  make[2]: Leaving directory `/home/postgres/InstallFile/postgresql-14.2/contrib/unaccent'  make -C vacuumlo install  make[2]: Entering directory `/home/postgres/InstallFile/postgresql-14.2/contrib/vacuumlo'  /usr/bin/mkdir -p '/applications/pgdb/app/pgsql/14.2/bin'  /usr/bin/install -c vacuumlo '/applications/pgdb/app/pgsql/14.2/bin'  make[2]: Leaving directory `/home/postgres/InstallFile/postgresql-14.2/contrib/vacuumlo'  make[1]: Leaving directory `/home/postgres/InstallFile/postgresql-14.2/contrib'  [postgres@dbtest03 postgresql-14.2]$        [postgres@dbtest03 ~]$ initdb --lc-collate=C --data-checksums     ## 클러스터 초기화  The files belonging to this database system will be owned by user "postgres".  This user must also own the server process.    The database cluster will be initialized with locales  COLLATE: C  CTYPE: ko\_KR.UTF-8  MESSAGES: ko\_KR.UTF-8  MONETARY: ko\_KR.UTF-8  NUMERIC: ko\_KR.UTF-8  TIME: ko\_KR.UTF-8  The default database encoding has accordingly been set to "UTF8".  initdb: could not find suitable text search configuration for locale "ko\_KR.UTF-8"  The default text search configuration will be set to "simple".    Data page checksums are enabled.    fixing permissions on existing directory /applications/pgdb/data ... ok  creating subdirectories ... ok  selecting dynamic shared memory implementation ... posix  selecting default max\_connections ... 100  selecting default shared\_buffers ... 128MB  selecting default time zone ... ROK  creating configuration files ... ok  running bootstrap script ... ok  performing post-bootstrap initialization ... ok  syncing data to disk ... ok    initdb: warning: enabling "trust" authentication for local connections  You can change this by editing pg\_hba.conf or using the option -A, or  --auth-local and --auth-host, the next time you run initdb.    Success. You can now start the database server using:    pg\_ctl -D /applications/pgdb/data -l logfile start    [postgres@dbtest03 ~]$ |

**시작 및 중지** 원본 접기

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23 | [postgres@dbtest03 data]$ pg\_ctl start     ## 시작  waiting for server to start....2022-03-18 15:50:01.478 KST [2017] LOG: starting PostgreSQL 14.2 on x86\_64-pc-linux-gnu, compiled by gcc (GCC) 4.8.5 20150623 (Red Hat 4.8.5-44), 64-bit  2022-03-18 15:50:01.478 KST [2017] LOG: listening on IPv6 address "::1", port 5432  2022-03-18 15:50:01.478 KST [2017] LOG: listening on IPv4 address "127.0.0.1", port 5432  2022-03-18 15:50:01.481 KST [2017] LOG: listening on Unix socket "/tmp/.s.PGSQL.5432"  2022-03-18 15:50:01.484 KST [2018] LOG: database system was shut down at 2022-03-18 15:49:19 KST  2022-03-18 15:50:01.486 KST [2017] LOG: database system is ready to accept connections  done  server started  [postgres@dbtest03 data]$  [postgres@dbtest03 ~]$ pg\_ctl stop     ## 중지  waiting for server to shut down....2022-03-18 16:19:02.273 KST [4682] LOG: received fast shutdown request  2022-03-18 16:19:02.274 KST [4682] LOG: aborting any active transactions  2022-03-18 16:19:02.275 KST [4682] LOG: background worker "logical replication launcher" (PID 4689) exited with exit code 1  2022-03-18 16:19:02.275 KST [4684] LOG: shutting down  2022-03-18 16:19:02.285 KST [4682] LOG: database system is shut down  done  server stopped  [postgres@dbtest03 ~]$  [postgres@dbtest03 ~]$ pg\_ctl status     ## 서비스 상태 확인  pg\_ctl: server is running (PID: 4944)  /applications/pgdb/app/pgsql/14.2/bin/postgres  [postgres@dbtest03 ~]$ |

**파라메타 설정 및 접근 설정** 원본 접기

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | 초기화  init 이전에 locale 설정 : # locale 확인 : export LANG=ko\_kr.UTF-8  initdb --lc-collate=C --data-checksums     ## 클러스터 초기화    접속 설정  파일 : /pgdb/data/pg\_hba.conf  host    all             all             0.0.0.0/0              md5  설정 변경후 : pg\_ctl reload  pg\_ctl stop and pg\_ctl start -w -c    파라메타 설정  파일 : /pgdb/data/postgresql.conf  https://pgtune.leopard.in.ua/ \* 해당 site 접속해서 정보 넣고 파라메타 확인하여 적용하기 |

3. 기타

**기타 설치** 원본 접기

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53 | ## plv8 Extension  yum install gcc-c++  yum install python3  yum install epel-release  yum install centos-release-scl-rh  yum install libcxx-devel     ## epel repo 필요  yum install bzip2  yum install llvm5.0 llvm-devel  yum install libstdc++-static.x86\_64  yum install gettext-devel  yum install openssl-devel  yum install autoconf  yum install glibc-static  yum install ninja-build  yum install kernel-devel    yum install python-devel  yum install llvm-toolset-7-clang  sudo yum install -y https://download.postgresql.org/pub/repos/yum/reporpms/EL-7-x86\_64/pgdg-redhat-repo-latest.noarch.rpm  yum install postgresql11-devel.x86\_64  yum install make  yum install lbzip2    ## 소스 파일 압축 해제 및 압축 해제 경로 이동  make PG\_CONFIG=/applications/pgdb/app/pgsql/11.15/bin/pg\_config     ## 소스파일 make (다소 시간 걸림)  make install PG\_CONFIG=/applications/pgdb/pgsql/11.15/bin/pg\_config     ## make install    ## 설치 후 $PGHOME/share/extension 아래 경로에 관련 파일 확인 가능  [postgres@gf-prd-dp-dev extension]$ ls -alh plv\*  -rw-r--r-- 1 postgres postgres  98  4월 12 13:58 plv8--1.5.0--2.3.9.sql  -rw-r--r-- 1 postgres postgres  98  4월 12 13:58 plv8--1.5.1--2.3.9.sql  -rw-r--r-- 1 postgres postgres  98  4월 12 13:58 plv8--1.5.2--2.3.9.sql  -rw-r--r-- 1 postgres postgres  98  4월 12 13:58 plv8--1.5.3--2.3.9.sql  -rw-r--r-- 1 postgres postgres  98  4월 12 13:58 plv8--1.5.4--2.3.9.sql  -rw-r--r-- 1 postgres postgres  98  4월 12 13:58 plv8--1.5.5--2.3.9.sql  -rw-r--r-- 1 postgres postgres  98  4월 12 13:58 plv8--1.5.6--2.3.9.sql  -rw-r--r-- 1 postgres postgres  98  4월 12 13:58 plv8--1.5.7--2.3.9.sql  -rw-r--r-- 1 postgres postgres  98  4월 12 13:58 plv8--2.0.0--2.3.9.sql  -rw-r--r-- 1 postgres postgres  98  4월 12 13:58 plv8--2.0.1--2.3.9.sql  -rw-r--r-- 1 postgres postgres  98  4월 12 13:58 plv8--2.0.3--2.3.9.sql  -rw-r--r-- 1 postgres postgres  98  4월 12 13:58 plv8--2.1.0--2.3.9.sql  -rw-r--r-- 1 postgres postgres  98  4월 12 13:58 plv8--2.3.0--2.3.9.sql  -rw-r--r-- 1 postgres postgres  98  4월 12 13:58 plv8--2.3.1--2.3.9.sql  -rw-r--r-- 1 postgres postgres  98  4월 12 13:58 plv8--2.3.2--2.3.9.sql  -rw-r--r-- 1 postgres postgres  98  4월 12 13:58 plv8--2.3.3--2.3.9.sql  -rw-r--r-- 1 postgres postgres  98  4월 12 13:58 plv8--2.3.4--2.3.9.sql  -rw-r--r-- 1 postgres postgres  98  4월 12 13:58 plv8--2.3.5--2.3.9.sql  -rw-r--r-- 1 postgres postgres  98  4월 12 13:58 plv8--2.3.6--2.3.9.sql  -rw-r--r-- 1 postgres postgres  98  4월 12 13:58 plv8--2.3.7--2.3.9.sql  -rw-r--r-- 1 postgres postgres  98  4월 12 13:58 plv8--2.3.8--2.3.9.sql  -rw-r--r-- 1 postgres postgres 658  4월 12 13:58 plv8--2.3.9.sql  -rw-r--r-- 1 postgres postgres 199  4월 12 13:58 plv8.control  [postgres@gf-prd-dp-dev extension]$ |